

A Novel and Computationally Efficient Joint Unscented Kalman Filtering Scheme for Parameter Estimation of a Class of Nonlinear Systems

ALTAN ONAT^{ID}

Electrical and Electronics Engineering Department, Engineering Faculty, Eskişehir Technical University, İki Eylül Campus, 26555 Eskişehir, Turkey

e-mail: altanonat@eskisehir.edu.tr

This work was supported by the Eskişehir Technical University.

ABSTRACT Unscented Kalman filter (UKF) is one type of the sigma point Kalman filters and it is based on unscented transformation. UKF is used for parameter estimation of various dynamic systems and for such purpose either joint UKF (JUKF) or dual UKF (DUKF) schemes are considered. JUKF is based on estimating states and parameters together by using only one filter. For DUKF, states and parameters are decoupled and two separate filters are considered. In this paper, a modification to standard JUKF is proposed for parameter estimation which is based on decoupling parameter vector and updating parameter estimates by considering the error transformation between measurements and transformed sigma points during measurement update into the parameter errors. A linear transformation is proposed for such a purpose. Thus, the computational complexity of the standard JUKF is reduced significantly since parameters are decoupled from the state vector while the convergence of parameter estimate(s) is guaranteed. The new modified JUKF scheme is promising to be used for the parameter estimation of dynamic systems for which a linear transformation between measurement and parameter errors can be obtained. The effectiveness of this new scheme is proven by applying it to two nonlinear dynamic systems.

INDEX TERMS Kalman filter, unscented Kalman filter, joint unscented Kalman filter, parameter estimation, dynamic systems.

I. INTRODUCTION

After the seminal paper by Kalman [1], intensive research on filtering and prediction problems by using this new method has gathered pace. In [2], a modification to Kalman filtering is proposed so that it can be applied to nonlinear systems, and later this method is called as extended Kalman filtering (EKF).

A viable alternative to EKF for filtering and estimation problems of nonlinear systems is UKF [3] and this method is based on UT [4]. UKF is a derivative-less Kalman filter for Gaussian approximate nonlinear estimation [5]. Thus, UKF eliminates the necessity of obtaining a Jacobian matrix during linearization of the system around the state. UKF and other similar types of Kalman filters are grouped as sigma point

Kalman filters (SPKF) by Van Der Merwe [5]. In a fairly recent study [6], the corrections to theoretical inconsistencies of discrete time unscented Kalman filter theory for state estimation is discussed and a tool is proposed to construct new UKFs in a consistent way.

UKF is not just used for state estimation, but also it is used for parameter estimation of nonlinear systems. Two approaches exist in case of parameter estimation. First one is the joint filtering approach and it is based on considering one filter for a state vector which includes both states and parameters. Second approach is based on using two simultaneous and separate filters for state and parameter estimation. In this approach, the output of the parameter estimator is the input of the state filter [5]. System illustrations for these approaches are given in the Fig. 1, 2 and x , y , θ represent system states, measurements and system parameters respectively. \hat{x}_k , $\hat{\theta}_k$ are state and parameter estimates, respectively.

The associate editor coordinating the review of this manuscript and approving it for publication was Kan Liu.

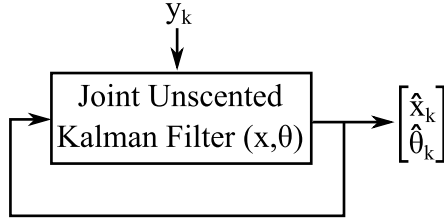


FIGURE 1. JUKF scheme, adapted from [5].

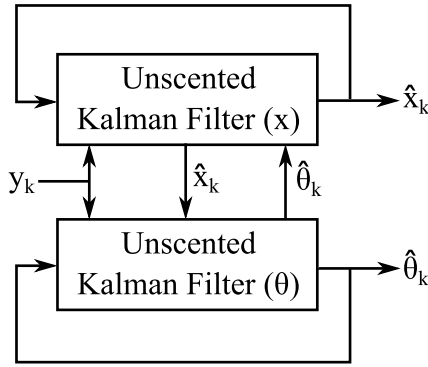


FIGURE 2. DUKF scheme, adapted from [5].

Fundamental difference between dual and joint filtering frameworks are explained in [5, p. 103]. In joint structure, states and parameters are joined together to create an augmented state vector and this allows calculation of the cross covariance between states and parameters as

$$E \left[\left(\hat{\tilde{x}} - E \left[\hat{\tilde{x}} \right] \right) \left(\hat{\tilde{x}} - E \left[\hat{\tilde{x}} \right] \right)^T \right] = \begin{bmatrix} \mathbf{P}_{x_k} & \mathbf{P}_{\theta_k x_k} \\ \mathbf{P}_{x_k \theta_k} & \mathbf{P}_{\theta_k} \end{bmatrix}, \quad (1)$$

where θ represents parameters, \mathbf{P}_{x_k} and $\mathbf{P}_{x_k \theta_k}$ are state covariance and cross covariance between states and parameters, respectively. On the contrary, dual filtering structure is based on separating state and parameter filter where in each filter state or parameter is assumed to be constant. As shown in [5], this separation in dual filtering causes $\mathbf{P}_{x_k \theta_k} = \mathbf{P}_{\theta_k x_k}^T = 0$, and this simply means that a cross covariance between states (i.e. measurements depending on the observability of states) and parameters is not calculated in dual filtering. Therefore, the joint structure is expected to be more effective than the dual filtering for parameter estimation [5].

JUKF has various application areas [7]–[15], it is robust and still considered as a parameter estimator in the literature. Majority of these applications are based on estimating a constant or slowly varying parameter for a mechanical, electrical, electromechanical system etc. with JUKF which is considered as fault diagnosis and condition monitoring method [16], [17]. Especially for such systems, computational complexity of JUKF can be high dependent on the number of estimated states and parameters. As stated previously, UKF is based on UT and after UT, $2L + 1$ sigma points are generated where L is the number of states. In JUKF, since parameter vector are concatenated to state vector, size of the state vector and the number of sigma points increases

dependent on the size of the parameter vector. As reported in [18], calculation of sigma points at each time update is the most computationally expensive part since it requires calculation of a matrix square root. In this work, instead of creating extra sigma points for constant or slowly changing parameters and calculating a full covariance matrix, a linear transformation, which transforms measurement errors into parameter error(s) at measurement update, is proposed. In this modified method, only standard UKF is considered for state estimation and parameters are estimated based on a linear transformation between measurements and parameters. Parameters are decoupled from the state vector and separate sigma points are constructed both for states and parameters during estimation and the covariance matrix is updated only for states. For example, assuming a dynamic system with 3 states and 2 parameters, states and parameters are concatenated in state vector and sigma point matrix propagated through nonlinear system functions has a size of 11×5 in standard JUKF, whereas in MJUKF, sigma points for states and parameters are created separately, they have the size of 7×3 , 7×2 , respectively and they are concatenated (7×5) during propagation through nonlinear system functions. Thus, a significant reduction in the computational complexity is provided with MJUKF since only 7 instances of nonlinear system functions are propagated, whereas it is 11 in standard JUKF. Additionally, a covariance matrix is not calculated for parameters during estimation and this also reduces the computational complexity in MJUKF. Main contribution of this new scheme is that it reduces the computational complexity of the standard JUKF while preserving parameter convergence. It should be noted that the use of this modification is applicable and promising for nonlinear dynamic systems for which a linear transformation between measurements and parameters can be obtained. Fortunately, such a linear transformation can be obtained in the majority of the previously stated applications on mechanical, electrical, electromechanical systems etc. for fault diagnosis and condition monitoring [12], [16], [17]. Furthermore, such approach also provides an indirect calculation of correlation between states and parameters. Therefore, this approach is expected to provide better results than the dual scheme and to have less computational complexity than joint scheme.

In the following section, details of the MJUKF is presented in comparison with the standard JUKF. Third section provides a simulation study which MJUKF is applied for parameter estimation in two nonlinear dynamic systems. It is shown that MJUKF provides promising results in terms of computational complexity with respect to the standard JUKF. The last section includes a discussion and conclusion about the new approach.

II. MODIFIED JOINT UNSCENTED KALMAN FILTER

UKF is based on UT and the main principle of UT is to create sample points which reflects the statistical properties of the system states instead of linearizing a nonlinear function. Considering a discrete time or discretized continuous time

dynamic system

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\theta}_{k-1}) + \mathbf{q}_k, \quad (2a)$$

$$\mathbf{y}_k = g(\mathbf{x}_k, \boldsymbol{\theta}_k) + \mathbf{r}_k \quad (2b)$$

where $\mathbf{x}_k = \mathbf{x}(kT_s)$ is the state vector with a *discrete time index* k and the sampling period T_s , \mathbf{u}_k is the input vector, $\boldsymbol{\theta}$ is the parameter vector. \mathbf{q} and \mathbf{r} are model and measurement noise vectors, respectively, f and g are the nonlinear functions for system states and measurements, respectively. In this study, the additive noise case is considered, where the model and measurement noise is purely additive, since it is often encountered [19].

Assuming that the state vector \mathbf{x} has dimension L and it has a mean $\hat{\mathbf{x}}$ and covariance \mathbf{P}_x , $2L + 1$ sigma points are generated during UT, which reflect the statistics of the system states. These sigma points are expressed as

$$\chi_{k-1} = [\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{x}}_{k-1} + \sqrt{(L+\lambda)\mathbf{P}_x}, \hat{\mathbf{x}}_{k-1} - \sqrt{(L+\lambda)\mathbf{P}_x}] \quad (3)$$

where λ is a scaling parameter and it is defined as

$$\lambda = \alpha^2 (L + \kappa) - L, \quad (4)$$

where α is for the scatter of sigma points around the mean $\hat{\mathbf{x}}$. It gets values between $10^{-4} \leq \alpha \leq 1$ [19]. κ is the secondary scaling parameter and it is usually chosen as $3 - L$ [19]. In JUKF, state vector \mathbf{x} incorporates the parameter vector $\boldsymbol{\theta}$. It should be noted that $\sqrt{(L+\lambda)\mathbf{P}_x}$ is matrix square root in JUKF. As stated in [18], generating sigma points at each update is the most computationally expensive part in UKF. In JUKF, since parameter vector is concatenated to state vector, computational complexity increases depending on the size of parameter vector. Hereby, the first important modification to standard JUKF is to create a separate sigma points for parameters and it is constructed as

$$\chi_{\theta} = \begin{bmatrix} \hat{\boldsymbol{\theta}}_1 \\ \hat{\boldsymbol{\theta}}_2 \\ \vdots \\ \hat{\boldsymbol{\theta}}_n \end{bmatrix}, \quad n = 2L + 1 \quad (5)$$

where scatter of parameters are determined *initially* as

$$\hat{\boldsymbol{\theta}}_i = (\hat{\boldsymbol{\theta}}_0 - \mathbf{p}_{\theta}(L+1)) + \mathbf{p}_{\theta}i, \quad i = 1, \dots, 2L + 1 \quad (6)$$

where L is the number of states in MJUKF, i is the index of the corresponding parameter sigma point, $\hat{\boldsymbol{\theta}}_0$ is the vector for initial parameter estimate, \mathbf{p}_{θ} is a row vector containing initial error covariance values for parameters. Thus, it determines the scatter of the parameters around the mean $\hat{\boldsymbol{\theta}}_0$ in parameter space. This step is just to determine an *initial sigma points* for parameter(s) which are distributed uniformly in parameter space. Since a matrix square root is not necessary to construct a sigma point matrix for parameter, this step is the first factor which reduces the computational complexity of the MJUKF with respect to the standard JUKF. In MJUKF, sigma points for states and parameters are concatenated and the new sigma

point vector is $\chi_{k-1}^{\dagger} = [\chi_{k-1} | \chi_{\theta}]$. In the next step, these new sigma points are propagated by using the nonlinear function

$$\chi_{k|k-1}^* = f(\chi_{k-1}^{\dagger}). \quad (7)$$

Now the mean $\hat{\mathbf{x}}_k^-$ (i.e. priori state estimate) and covariance of priori sigma points $\chi_{k|k-1}^*$ are given as

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \chi_{i,k|k-1}^*, \quad (8)$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} (\chi_{i,k|k-1}^* - \hat{\mathbf{x}}_k^-)(\chi_{i,k|k-1}^* - \hat{\mathbf{x}}_k^-)^T + \mathbf{Q}, \quad (9)$$

where \mathbf{Q} is the process noise covariance, $W_i^{(m)}$ and $W_i^{(c)}$ are weights for mean and covariance and given as

$$W_0^{(m)} = \frac{\lambda}{L + \lambda}, \quad (10a)$$

$$W_0^{(c)} = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta), \quad (10b)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L + \lambda)}, \quad i = 1, \dots, 2L + 1 \quad (10c)$$

where β is a parameter for prior knowledge of the state distribution and set optimally to 2 for Gaussian distributions [20].

These propagated sigma points are also updated for the measurement sigma point

$$\Upsilon_{k|k-1}^* = g(\chi_{k|k-1}^*). \quad (11)$$

Then, mean $\hat{\mathbf{y}}_k^-$ (i.e. priori measurement estimate) is expressed as

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \Upsilon_{i,k|k-1}^*. \quad (12)$$

Cross covariances are given as

$$\mathbf{P}_{\mathbf{y}_k \mathbf{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} (\Upsilon_{i,k|k-1}^* - \hat{\mathbf{y}}_k^-)(\Upsilon_{i,k|k-1}^* - \hat{\mathbf{y}}_k^-)^T + \mathbf{R}, \quad (13)$$

$$\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} (\chi_{i,k|k-1}^* - \hat{\mathbf{x}}_k^-)(\Upsilon_{i,k|k-1}^* - \hat{\mathbf{y}}_k^-)^T. \quad (14)$$

where \mathbf{R} is the measurement noise covariance. The final step in standard JUKF is to calculate Kalman gain, state estimate and update state covariance. These are

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} \mathbf{P}_{\mathbf{y}_k \mathbf{y}_k}^{-1}, \quad (15)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_k^-), \quad (16)$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{\mathbf{y}_k \mathbf{y}_k} \mathbf{K}_k^T. \quad (17)$$

So far, however, except creating separate sigma points for parameters and concatenating them with sigma points for states, given procedure does not differ from JUKF. In the Kalman filter framework, following state space model can be used for the parameter estimation

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \mathbf{d}_k, \quad (18)$$

where the equation above is a stationary process driven by the noise \mathbf{d}_k and the output is already defined in (2). As stated in [5, p. 89], by using the Kalman filtering framework for parameter estimation following *prediction error* is minimized

$$J(\theta_k) = \sum_{t=1}^k [\mathbf{y}_k - g(\mathbf{x}_k, \theta_k)] (\mathbf{R}_e^{-1}) [\mathbf{y}_k - g(\mathbf{x}_k, \theta_k)]^T. \quad (19)$$

\mathbf{R}_e is a constant diagonal matrix and it can be set arbitrarily [5], e.g. $\mathbf{R}_e = \frac{1}{2} \mathbf{I}$ where \mathbf{I} is the identity matrix. Nevertheless, innovations covariance $E[\mathbf{d}_k \mathbf{d}_k^T] = \mathbf{R}_{d_k}$ is critical for the convergence rate and the performance of the estimator. Three options are specified in [5] for the selection of \mathbf{R}_{d_k} , and one of the options is to select arbitrary fixed diagonal value and then it can be annealed toward zero by using the Kalman filtering framework. In this study, the method, namely MJUKF, seeks a parameter update rule based on this fact, instead of using a second filter to anneal this innovation covariance towards zero. Therefore, for the dynamic systems in which there is a linear relationship between measurements and parameters, \mathbf{R}_{d_k} can be selected as the linearly transformed error covariance between measurements and propagated sigma points through the measurement function $g(\mathbf{x}_k, \theta_k)$. In MJUKF, parameter update rule presented in next equation for parameter sigma points is selected and this step is the second important modification to the standard JUKF. This rule is defined as

$$\mathbf{x}_{\theta_{i,k}} = \hat{\theta}_{k-1} - \xi T \left(\mathbf{y}_k - \Upsilon_{i,k|k-1}^* \right), \quad i = 1, \dots, 2L + 1 \quad (20)$$

where \mathbf{x}_{θ} is defined in (5) and contains parameter sigma points $\theta_{i,k}$ at k^{th} discrete time index where $i = 1, \dots, 2L + 1$, and $\hat{\theta}_{k-1}$ is the parameter estimate at $(k - 1)^{th}$ discrete time index, L is the number of states to be estimated, T is a transformation matrix which transforms measurement errors of measurement sigma point $\Upsilon_{i,k|k-1}^*$ at k^{th} discrete time index to parameter errors, ξ is a scaling parameter for the transformation. T is a $L_{\theta} \times L_m$ matrix, where L_{θ} , L_m is the number of estimated parameters and measurements, respectively. It is clear from the (20) that selection of $\mathbf{d}_k = \xi T \left(\mathbf{y}_k - \Upsilon_{i,k|k-1}^* \right)$ can anneal $E[\mathbf{d}_k \mathbf{d}_k^T] = \mathbf{R}_{d_k}$ to zero, if a linear transformation between measurement and parameter error is obtained. Parameter estimate at k^{th} discrete time index is simply the average of parameter sigma points and given as

$$\hat{\theta}_k = \frac{1}{2L + 1} \sum_{i=0}^{2L} \mathbf{x}_{\theta_i}. \quad (21)$$

In order to analyze how the modified sigma point vector brings computational complexity reduction, a system with L number of states and L_{θ} number of parameters is considered. Supposing that if all parameters are known in the system, then the number of generated sigma points is $2L + 1$ and the size of the sigma point matrix is $(2L + 1) \times L$, which corresponds to the case in standard UKF. In case of JUKF, the state and

parameter vectors are concatenated and the number of generated sigma points is $2(L + L_{\theta}) + 1$ and the size of the sigma point matrix is $(2(L + L_{\theta}) + 1) \times (L + L_{\theta})$. For the nonlinear systems, in which the parameters are *constant unknowns*, *linear in measurements and uncorrelated with measurements*, $2L_{\theta}$ number of *extra* sigma points are generated. Structure of the sigma point matrix can be shown for JUKF as

$$\chi_{JUKF} = \begin{bmatrix} \chi_{(2L+1) \times L} & \chi_{(2L+1) \times L_{\theta}} \\ \chi_{(2L_{\theta}) \times L} & \chi_{(2L_{\theta}) \times L_{\theta}} \end{bmatrix}. \quad (22)$$

It can be seen in the previous equation that the concatenation of parameter vector results in generation of $2L_{\theta}$ extra sigma points in JUKF for the constant unknowns. In MJUKF, the sigma points generated due to the constant parameters are omitted and just the concatenation of state sigma point matrix with parameter sigma point matrix is considered as

$$\chi_{MJUKF} = [\chi_{(2L+1) \times L} | \chi_{(2L+1) \times L_{\theta}}]. \quad (23)$$

In conclusion, the reduction of the number of sigma points in percentage equals to $\frac{(2L_{\theta})}{(2(L+L_{\theta})+1)} \times 100$. Nevertheless, since the parameter update rules are different in MJUKF and JUKF, actual reduction in computational complexity for MJUKF is lower than the reduction of the number of sigma points in percentage. The main reason is that the separate calculation of parameter sigma points in (20) and parameter update (21) in MJUKF. In the following section, it is shown for the given Van der Pol Oscillator and Lorenz 63 system that reduction in elapsed times for the simulations is approximately ($\approx 10\%$) lower than the reduction in percentage for number of sigma points. For example in Van der Pol oscillator, there are 2 states and 1 parameters. The reduction of sigma points in MJUKF is $\frac{(2L_{\theta})}{(2(L+L_{\theta})+1)} \times 100 = \frac{(2 \times 1)}{(2(2+1)+1)} \times 100 \approx 28.5\%$, whereas in Lorenz 63 system there are 3 states and 3 parameters and the reduction of sigma points in MJUKF is $\frac{(2L_{\theta})}{(2(L+L_{\theta})+1)} \times 100 = \frac{(2 \times 3)}{(2(3+3)+1)} \times 100 \approx 46\%$. However, the actual reduction in average elapsed times are $\approx 20\%$ and $\approx 36\%$ for Van der Pol oscillator and Lorenz 63 system, respectively. The reduction of sigma points is considered as an indicator to reveal the computational complexity of MJUKF. In [21], the computation cost of the JUKF is expressed as $\mathcal{O}((L + L_{\theta})^3)$ and it is referenced to [5] in [21] that the computation cost of the standard UKF is $\mathcal{O}(L^3)$. In \mathcal{O} notation, clearly computation cost of MJUKF is *close* to the $\mathcal{O}(L^3)$. Nevertheless, since MJUKF is considered as a *parameter estimator* as well, its computation cost is slightly higher than the standard UKF which is a state estimator with $\mathcal{O}(L^3)$.

In standard sigma point Kalman filters (e.g. JUKF, DUKF), it is assumed that the prior knowledge on system states and parameters are known. This is a drawback of the family of standard sigma point Kalman filters and in order to overcome this limitation adaptive versions of unscented Kalman filter is proposed based on *maximum likelihood estimation* (i.e. MLE) [22]. Since MJUKF is a sigma point Kalman filter with a modification to standard JUKF, it is assumed here that prior statistics for states (i.e. \mathbf{P}_x) and parameters (i.e. \mathbf{p}_{θ}) are available. Therefore, *maximum a-posteriori estimation*

perspective (i.e. MAP), which is a *regularization* of MLE in case of prior statistics, is considered here along with the *statistical linearization* (i.e. SL) to prove that the selection of parameter update rule in (20) provides an extremum for the *maximum posterior likelihood equation* (52), and it is used for the theoretical analysis of parameter estimation with sigma point Kalman filters in [5] and also in this study. *Maximum a-posteriori estimation* perspective is also considered in [21], but it is proven in [21] that the proposed new DUKF is equivalent to the cost function of standard JUKF. Considering the equations for parameter estimation in (2b) and (18) posterior distribution of the system parameters

$$p(\theta_k | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_{1:k} | \theta_k) p(\theta_k)}{p(\mathbf{y}_{1:k})}. \quad (24)$$

This can be further expanded

$$p(\theta_k | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k, \mathbf{y}_{1:k-1} | \theta_k) p(\theta_k)}{p(\mathbf{y}_{1:k})}, \quad (25)$$

$$p(\theta_k | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k | \mathbf{y}_{1:k-1}, \theta_k) p(\mathbf{y}_{1:k-1} | \theta_k) p(\theta_k)}{p(\mathbf{y}_{1:k})}, \quad (26)$$

$$p(\theta_k | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k | \theta_k) p(\mathbf{y}_{1:k-1} | \theta_k) p(\theta_k) p(\mathbf{y}_{1:k-1})}{p(\mathbf{y}_{1:k}) p(\mathbf{y}_{1:k-1})}, \quad (27)$$

$$p(\theta_k | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k | \theta_k) p(\theta_k | \mathbf{y}_{1:k-1})}{p(\mathbf{y}_{1:k}) / p(\mathbf{y}_{1:k-1})}. \quad (28)$$

The last equation above is obtained by using the Bayes rule and the conditional independence of the observation for the current state $[p(\mathbf{y}_k | \mathbf{y}_{k-1}, \theta_k) \doteq p(\mathbf{y}_k | \theta_k)]$, and also both numerator and denominator are multiplied by $p(\mathbf{y}_{1:k-1})$ going from (27) to (28). In (28), the θ_k which maximizes the following equation is selected as MAP parameter estimate

$$\hat{\theta}_k^{MAP} = \arg \max_{\theta_k} [p(\mathbf{y}_k | \theta_k) p(\theta_k | \mathbf{y}_{1:k-1})]. \quad (29)$$

If (29) is written as minimization of the negative logarithmic function of the right-hand side

$$\hat{\theta}_k^{MAP} = \arg \min [-\ln(p(\mathbf{y}_k | \theta_k) p(\theta_k | \mathbf{y}_{1:k-1}))], \quad (30)$$

$$\hat{\theta}_k^{MAP} = \arg \min [-\ln(p(\mathbf{y}_k | \theta_k)) - \ln(p(\theta_k | \mathbf{y}_{1:k-1}))], \quad (31)$$

$$\hat{\theta}_k^{MAP} = \arg \min [J(\theta_k)], \quad (32)$$

where $J(\theta_k) = -\ln(p(\mathbf{y}_k | \theta_k)) - \ln(p(\theta_k | \mathbf{y}_{1:k-1}))$ is called *posterior log-likelihood function* [5], [21]. For Kalman filtering framework, the main assumption is that all densities are Gaussian. Sigma point Kalman filters (e.g. JUKF or MJUKF) are not an exception and densities for the system given for $J(\hat{\theta}_k)$ are expressed as

$$p(\theta_k | \mathbf{y}_{1:k-1}) = \frac{1}{\sqrt{(2\pi)^{L_\theta} |\mathbf{P}_{\hat{\theta}_k}^-|}} \times \exp \left[-\frac{1}{2} (\theta_k - \hat{\theta}_k^-)^T \mathbf{P}_{\hat{\theta}_k}^- (\theta_k - \hat{\theta}_k^-) \right], \quad (33)$$

where $\hat{\theta}_k^-$ is the prior parameter estimate, $\mathbf{P}_{\hat{\theta}_k}^-$ is its covariance and L_θ is the dimension of the parameter vector. Likewise,

$$p(\mathbf{y}_k | \theta_k) = \frac{1}{\sqrt{(2\pi)^{L_m} |\mathbf{R}_e|}} \times \exp \left[-\frac{1}{2} (\mathbf{y}_k - g(\mathbf{x}_k, \theta_k))^T (\mathbf{R}_e)^{-1} (\mathbf{y}_k - g(\mathbf{x}_k, \theta_k)) \right], \quad (34)$$

where $g(\mathbf{x}_k, \theta_k)$ is nonlinear observation function given in (2b), \mathbf{R}_e is the observation noise covariance and L_m is the dimension of the observation function. The nonlinear observation model presented here is transformed into a *statistically linearized* form as it is carried out for sigma point approach (e.g. JUKF) in [5]. Furthermore, it is assumed here that there is a linear relationship between measurement and parameter errors. A linear approximation of the measurement function (2b) around the state \mathbf{x}_k is desired with respect to the estimated parameter vector which is a random variable and it is given as

$$\mathbf{y} = g(\mathbf{x}_k, \theta_k) \approx \mathbf{A}\theta_k + \mathbf{b}. \quad (35)$$

In the last equation, the aim is to find matrix \mathbf{A} and vector \mathbf{b} such that distribution of parameters is considered and obtaining a linearization of the measurement function in statistical sense. The approximation error is given as

$$\epsilon_k \doteq g(\mathbf{x}_k, \theta_k) - \mathbf{A}\theta_k + \mathbf{b}. \quad (36)$$

It is appropriate to choose \mathbf{A} and \mathbf{b} such that $J = E[\epsilon_k^T \mathbf{W} \epsilon_k]$ is minimized for a positive semi-definite matrix \mathbf{W} . Substituting (36) in J and taking the partial derivative of this expression with respect to \mathbf{b}

$$E[\mathbf{W}(g(\mathbf{x}_k, \theta_k) - \mathbf{A}\theta_k - \mathbf{b})] = 0, \quad (37)$$

so \mathbf{b} minimizing J can be found as

$$\mathbf{b} = E[(g(\mathbf{x}_k, \theta_k)) - E[\mathbf{A}\theta_k]] = 0, \quad (38)$$

$$\mathbf{b} = \bar{\mathbf{y}}_k - \mathbf{A}\bar{\theta}_k. \quad (39)$$

In parameter estimation with Kalman filtering framework, last equation for \mathbf{b} can be written as

$$\mathbf{b} = \hat{\mathbf{y}}_k - \mathbf{A}\hat{\theta}_k. \quad (40)$$

Similarly, substituting (39) in J and taking the partial derivative of this expression with respect to \mathbf{A}

$$E[\mathbf{W}[\mathbf{A}\tilde{\theta}_k\tilde{\theta}_k^T + (g(\mathbf{x}_k, \theta_k) - \bar{\mathbf{y}})\tilde{\theta}_k^T]] = 0, \quad (41)$$

where $\tilde{\theta}_k = \theta_k - \bar{\theta}_k$ and the \mathbf{A} minimizing J is

$$\mathbf{A} = E[(g(\mathbf{x}_k, \theta_k) - \bar{\mathbf{y}}_k)\tilde{\theta}_k^T] E[\tilde{\theta}_k\tilde{\theta}_k^T]^{-1}, \quad (42)$$

$$\mathbf{A} = E[(\theta_k - \bar{\theta}_k)(\mathbf{y}_k - \bar{\mathbf{y}}_k)^T]^T E[(\theta_k - \bar{\theta}_k)(\theta_k - \bar{\theta}_k)^T]^{-1}, \quad (43)$$

$$\mathbf{A} = \mathbf{P}_{\theta_k \mathbf{y}_k}^T \mathbf{P}_{\theta_k}^{-1}, \quad (44)$$

where \mathbf{P}_{θ_k} is the covariance matrix of θ_k and $\mathbf{P}_{\theta_k \mathbf{y}_k}$ is the cross covariance matrix between θ_k and $\mathbf{y} = g(\mathbf{x}_k, \theta_k)$. Here both \mathbf{b} and \mathbf{A} are independent of \mathbf{W} and they provide a generalized minimum mean square error linearized approximation of nonlinear observation function. This linear approximation will be substituted in (34), and for this purpose, (35) can be rewritten by using (40) as

$$g(\mathbf{x}_k, \theta_k) = \mathbf{A}\theta_k + \mathbf{b} + \epsilon_k, \quad (45)$$

$$g(\mathbf{x}_k, \theta_k) = \mathbf{A}\theta_k + \hat{\mathbf{y}}_k^- - \mathbf{A}\hat{\theta}_k^- + \epsilon_k, \quad (46)$$

$$g(\mathbf{x}_k, \theta_k) = \mathbf{A}(\theta_k - \hat{\theta}_k^-) + \hat{\mathbf{y}}_k^- + \epsilon_k, \quad (47)$$

where ϵ_k is the statistical linearization error and it is a Gaussian random variable with covariance \mathbf{P}_ϵ . Substituting (47) into (2b)

$$\mathbf{y}_k = \mathbf{A}(\theta_k - \hat{\theta}_k^-) + \hat{\mathbf{y}}_k^- + \epsilon_k + \mathbf{r}_k, \quad (48)$$

in the last equation the term $\tilde{\epsilon}_k = \epsilon_k + \mathbf{r}_k$ is named as *effective observation noise* and since both ϵ_k and \mathbf{r}_k are Gaussian random variable, their sum is also a Gaussian random variable with covariance

$$\mathbf{R}_{\tilde{\epsilon}} = \mathbf{P}_\epsilon + \mathbf{R}_e, \quad (49)$$

where \mathbf{R}_e is the measurement noise covariance defined as $\mathbf{R}_e = E[\mathbf{r}_k \mathbf{r}_k^T]$ and \mathbf{r}_k is presented in (2b). If the alternative form in (48) with the term $\tilde{\epsilon}_k = \epsilon_k + \mathbf{r}_k$ is substituted in (34), then measurement likelihood density function is presented as

$$p(\mathbf{y}_k | \theta_k) = \frac{1}{\sqrt{(2\pi)^{L_m} |\mathbf{R}_e|}} \exp \left[-\frac{1}{2} \left(\mathbf{y}_k - \mathbf{A}(\theta_k - \hat{\theta}_k^-) - \hat{\mathbf{y}}_k^- \right)^T (\mathbf{R}_{\tilde{\epsilon}})^{-1} \left(\mathbf{y}_k - \mathbf{A}(\theta_k - \hat{\theta}_k^-) - \hat{\mathbf{y}}_k^- \right) \right]. \quad (50)$$

Now posterior log-likelihood function $J(\theta_k) = -\ln(p(\mathbf{y}_k | \theta_k)) - \ln(p(\theta_k | \mathbf{y}_{1:k-1}))$ can be rewritten by replacing (33) and (50) into posterior log-likelihood function

$$J(\theta_k) = \frac{1}{2} \left[\mathbf{y}_k - \mathbf{A}(\theta_k - \hat{\theta}_k^-) - \hat{\mathbf{y}}_k^- \right]^T (\mathbf{R}_{\tilde{\epsilon}})^{-1} \left[\mathbf{y}_k - \mathbf{A}(\theta_k - \hat{\theta}_k^-) - \hat{\mathbf{y}}_k^- \right] + \frac{1}{2} (\theta_k - \hat{\theta}_k^-)^T (\mathbf{P}_{\hat{\theta}_k}^-)^{-1} (\theta_k - \hat{\theta}_k^-). \quad (51)$$

The MAP parameter estimate is simply found by substituting (51) into (32) and taking the partial derivative of this expression with respect to θ_k to obtain extremum of posterior log likelihood function. This differential condition is defined as

$$\frac{\partial}{\partial \theta_k} J(\hat{\theta}_k^{MAP}) = 0, \quad (52)$$

and it is named as *maximum posterior likelihood function*. Taking the derivative of (51) with respect to θ_k and equating it to zero provides the following

$$\mathbf{P}_{\theta_k}^{-1} (\hat{\theta}_k^{MAP} - \hat{\theta}_k^-) - \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} \left[(\mathbf{y}_k - \mathbf{A}(\hat{\theta}_k^{MAP} - \hat{\theta}_k^-) - \hat{\mathbf{y}}_k^-) \right] = 0. \quad (53)$$

If (53) is solved for $\hat{\theta}_k^{MAP}$:

$$\mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} \left[(\mathbf{y}_k - \mathbf{A}(\hat{\theta}_k^{MAP} - \hat{\theta}_k^-) - \hat{\mathbf{y}}_k^-) \right] = \mathbf{P}_{\theta_k}^{-1} (\hat{\theta}_k^{MAP} - \hat{\theta}_k^-), \quad (54)$$

$$\left[\mathbf{P}_{\theta_k}^{-1} + \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} \mathbf{A} \right] (\hat{\theta}_k^{MAP} - \hat{\theta}_k^-) = \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} (\mathbf{y}_k - \hat{\mathbf{y}}_k^-), \quad (55)$$

$$\hat{\theta}_k^{MAP} - \hat{\theta}_k^- = \left[\mathbf{P}_{\theta_k}^{-1} + \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} \mathbf{A} \right]^{-1} \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} (\mathbf{y}_k - \hat{\mathbf{y}}_k^-), \quad (56)$$

$$\hat{\theta}_k^{MAP} - \hat{\theta}_k^- = \left[\mathbf{P}_{\theta_k}^{-1} + \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} \mathbf{A} \right]^{-1} \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} (\mathbf{y}_k - \hat{\mathbf{y}}_k^-), \quad (57)$$

$$\hat{\theta}_k^{MAP} = \hat{\theta}_k^- + \left[\mathbf{P}_{\theta_k}^{-1} + \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} \mathbf{A} \right]^{-1} \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} (\mathbf{y}_k - \hat{\mathbf{y}}_k^-). \quad (58)$$

(58) represents the maximum a posteriori estimate of $\hat{\theta}_k$. It has been already shown in [5] that the selection Kalman gain in sigma point Kalman filters for parameter estimation both in JUKF and DUKF

$$\mathbf{K} = \left[\mathbf{P}_{\theta_k}^{-1} + \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} \mathbf{A} \right]^{-1} \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1}, \quad (59)$$

results in the standard measurement update for the state, i.e.

$$\hat{\theta}_k^{MAP} = \hat{\theta}_k^- + \mathbf{K}(\mathbf{y}_k - \hat{\mathbf{y}}_k^-). \quad (60)$$

In JUKF, Kalman gain is updated with the rule $\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} \mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^{-1}$, and in DUKF, it is updated with $\mathbf{K}_k = \mathbf{P}_{\theta_k \mathbf{y}_k} \mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^{-1}$. In JUKF and DUKF, $\mathbf{R}_{\tilde{\epsilon}}^{-1}$ is selected constant so that a variable Kalman gain \mathbf{K}_k is considered by updating covariance matrices $\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k}$, $\mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}$, $\mathbf{P}_{\theta_k \mathbf{y}_k}$. Whereas in MJUKF, a constant Kalman gain is proposed for the update rule by selecting a variable $\mathbf{R}_{\tilde{\epsilon}}^{-1}$, so that *no covariance calculation* is necessary in MJUKF. However, still parameter update rule (20) in MJUKF satisfies the condition given in (58) for *maximum a-posteriori* parameter estimate. If equations for both MAP parameter estimate and the update rule in MJUKF are written together

$$\chi_{\theta_{i,k}} = \hat{\theta}_{k-1} - \xi T (\mathbf{y}_k - \Upsilon_{i,k|k-1}^*), \quad i = 1, \dots, 2L + 1 \quad (61)$$

$$\hat{\theta}_k^{MAP} = \hat{\theta}_k^- + \left[\mathbf{P}_{\theta_k}^{-1} + \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} \mathbf{A} \right]^{-1} \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} (\mathbf{y}_k - \hat{\mathbf{y}}_k^-), \quad (62)$$

it is clear that each parameter estimate vector in parameter sigma point matrix $\chi_{\theta_{i,k}}$ is a MAP parameter estimate.

It should be noted that $\hat{\theta}_{k-1} = \hat{\theta}_k^-$ and $\Upsilon_{i,k|k-1}^*$ (i.e. propagation of each sigma point through the nonlinear observation function expressed in (11)) represents each measurement estimate (i.e. $\hat{\mathbf{y}}_k^-$) for corresponding sigma point. Obviously, if (61) and (62) are compared, it is apparent to see that

$$\left[\mathbf{P}_{\theta_k}^{-1} + \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} \mathbf{A} \right]^{-1} \mathbf{A}^T \mathbf{R}_{\tilde{\epsilon}}^{-1} = -\xi T. \quad (63)$$

(63) is rearranged to find $\mathbf{R}_{\tilde{z}}$,

$$\mathbf{R}_{\tilde{z}} = \left[-\xi \left(\mathbf{A}\mathbf{A}^T \right)^{-1} \mathbf{A}\mathbf{P}_{\theta_k}^{-1} T [I + \xi \mathbf{A}T]^{-1} \right]^{-1}, \quad (64)$$

where I is an identity matrix. Selection of $\mathbf{R}_{\tilde{z}}$ as expressed in (64) implies that the Kalman gain in MJUKF is constant and equal to $\mathbf{K} = -\xi T$ so that each sigma point for parameter estimate is a MAP estimate and each sigma point satisfy the condition given in (58). This proves that MJUKF is a sigma point Kalman filter and each parameter sigma point in MJUKF is a *maximum a-posteriori* parameter estimate. MJUKF reduces computational complexity since there is no covariance calculation as it is in JUKF and DUKF, it does not require a second filter and it uses less number of sigma points for parameter estimation compared to JUKF.

In the selection of T , a weighted measurement error approach can be used. Simply, T can be selected based on the *transpose* of linear transformation matrix \mathbf{A} presented in (35), (42), (43), (44). Two methods are proposed to obtain the transformation matrix T . First method is based on obtaining a Jacobian matrix (i.e. \mathbf{A}) for measurements with respect to estimated parameters. It is apparent that a Jacobian matrix can be obtained if estimated parameters explicitly appear in the nonlinear state and measurement functions presented in (2). Let the number of estimated parameters be L_θ , and the number of measurements L_m , then the Jacobian matrix is expressed as

$$\mathbf{J}(\theta_1, \theta_2, \dots, \theta_n) = \begin{bmatrix} \frac{\partial g_1}{\partial \theta_1} & \cdots & \frac{\partial g_1}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial \theta_1} & \cdots & \frac{\partial g_m}{\partial \theta_n} \end{bmatrix}, \quad (65)$$

where g_i $i = 1, \dots, m$ is the nonlinear function for the corresponding measurement presented in (2), index m represents the number of measurements (i.e. L_m) and index n represents the number of parameters (i.e. L_θ). However, nonlinear function g for measurement cannot be obtained in terms of states and parameters for some dynamic systems in which measurements are *completely equal to the states*. In this case, *nonlinear function f for the system states* can be considered for the determination of the transformation matrix T and it is explained in the following section with an example.

Obviously, transformation matrix can be obtained after Jacobian matrix is determined. This matrix simply indicates the rate change of the measurements with respect to parameters and it is determined based on the transpose of the Jacobian. Thus, elements of the transformation matrix are determined based on the Jacobian matrix. The basic intuition about T is that the sum of elements in each column must be equal to unity. This sum is mathematically expressed as

$$T = \begin{bmatrix} t_{11} & \cdots & t_{1m} \\ \vdots & \ddots & \vdots \\ t_{n1} & \cdots & t_{nm} \end{bmatrix}, \quad (66)$$

$$\sum_{j=1}^m t_{ij} = 1, \quad i = 1, \dots, n \quad (67)$$

where t_{ij} represents the each element of the transformation matrix.

Additionally, the element for the corresponding measurement in the related column of T , which is influenced by the corresponding parameter, is chosen intentionally higher than elements of other measurements which are not influenced by the parameter. Therefore, after obtaining the Jacobian presented in (65), the rows of the T can be selected based on the columns of the Jacobian matrix. The intuition given in (67) is not strict since selection of the elements of T can be adjusted by scaling parameter ξ . The important point is that the ratio of influential measurements for parameter estimates for the corresponding element of the transformation matrix T must be chosen higher than the elements of the other measurements.

Nevertheless, estimated parameters do not explicitly appear in system equations for some dynamic systems. In such cases, other method to obtain a transformation matrix is to consider simulations of the considered system and determine the influence of each parameter on measurements. Both methods to determine transformation matrix is explained in the first example of the following section.

ξ is also a critical parameter for this modified approach which scales the measurement errors to parameter errors. ξ is similar to the parameter α in standard JUKF, and it simply controls the difference between measurement sigma points and the measurements. The basic intuition to select ξ is similar to the determination of α , and depending on the problem, α is selected a small value $10^{-4} \leq \alpha \leq 1$ as stated in [5] and [23]. It should be noted that value of the ξ is strongly dependent on the noise level of the measurements since a separate covariance matrix is not calculated for estimated parameters. Therefore, ξ can be different for the same system with different noise levels unlike α . Furthermore, ξ controls the sign in (20) so that ξ can be negative depending on the problem. In this study, a small value for ξ is proposed and the limits for this parameter is determined as $10^{-4} \leq |\xi| \leq 1$ similar to the α in standard JUKF. Additionally, another function of this parameter is that it determines the convergence rate of the parameter estimation and it is revealed in following section. The algorithmic approach for MJUKF is also presented in Algorithm 1.

III. APPLICATION OF MODIFIED JOINT UNSCENTED KALMAN FILTER

Similar to the assumptions given in [24], measurement update and sampling frequency are same. Besides, system model and estimator model are same. The filter parameters are selected as $\alpha = 1$, $\beta = 2$ and $\kappa = 0$. In all simulations, a fourth order Runge–Kutta integration method is used and all codes are written in MATLAB software.

Algorithm 1 MJUKF Algorithm for Additive Noise Case Algorithm

```

1: Define Filter Parameters
2:    $L, L_m, L_\theta$  ▷ Size of state, measurement and parameter vector, respectively
3:    $\alpha \leftarrow 1$  ▷ ( $10^{-4} \leq \alpha \leq 1$ )
4:    $\kappa \leftarrow 0$  ▷ (generally  $\kappa = 3 - L$ )
5:    $\beta \leftarrow 2$ 
6:    $\lambda \leftarrow \alpha^2(L + \kappa) - L$ 
7:    $W_0^{(m)} \leftarrow \frac{\lambda}{L + \lambda}$ 
8:    $W_0^{(c)} \leftarrow \frac{\lambda}{L + \lambda} + 1 - \alpha^2 + \beta$ 
9:   for  $i \leftarrow \{1, \dots, 2L\}$  do
10:     $W_i^{(m)} \leftarrow W_i^{(c)} := \frac{1}{2(L + \lambda)}$ 
11:   end for
12: End
13: function UKF( $\hat{\mathbf{x}}, \mathbf{P}_x, \mathbf{p}_\theta$ )
14:   Initialize
15:     $\hat{\mathbf{x}}_0 \leftarrow \mathbb{E}[\mathbf{x}_0]$ 
16:     $\mathbf{P}_0 \leftarrow \mathbb{E}[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]$ 
17:     $\hat{\boldsymbol{\theta}}_0 \leftarrow \mathbb{E}[\boldsymbol{\theta}_0]$ 
18:     $\mathbf{p}_\theta \leftarrow \mathbb{E}[(\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}_0)(\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}_0)^T]$ 
19:    for  $i \in \{1, \dots, n = 2L + 1\}$  do
20:      $\hat{\boldsymbol{\theta}}_i \leftarrow (\hat{\boldsymbol{\theta}}_0 - \mathbf{p}_\theta(L + 1)) + \mathbf{p}_\theta i$ ,
21:    end for
22:     $\boldsymbol{\chi}_\theta \leftarrow [\hat{\boldsymbol{\theta}}_1 \ \hat{\boldsymbol{\theta}}_2 \ \dots \ \hat{\boldsymbol{\theta}}_n]^T$ 
23:   End
24:   for  $k \in \{1, \dots, \infty\}$  do
25:    function Sigma Points( $\hat{\mathbf{x}}, \mathbf{P}_x$ )
26:      $\boldsymbol{\chi}_{k-1} \leftarrow [\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{x}}_{k-1} + \sqrt{(L + \lambda)\mathbf{P}_x}, \hat{\mathbf{x}}_{k-1} - \sqrt{(L + \lambda)\mathbf{P}_x}]$ 
27:    end function
28:    function Time Update( $\boldsymbol{\chi}_{k-1}, \mathbf{P}, \mathbf{Q}, \boldsymbol{\chi}_\theta$ )
29:      $\boldsymbol{\chi}_{k-1}^\dagger \leftarrow [\boldsymbol{\chi}_{k-1} | \boldsymbol{\chi}_\theta]$ 
30:      $\boldsymbol{\chi}_{k|k-1}^* \leftarrow f(\boldsymbol{\chi}_{k-1}^\dagger)$ 
31:      $\hat{\mathbf{x}}_k^- \leftarrow \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{\chi}_{i,k|k-1}^*$ 
32:      $\mathbf{P}_k^- \leftarrow \sum_{i=0}^{2L} W_i^{(c)} (\boldsymbol{\chi}_{i,k|k-1}^* - \hat{\mathbf{x}}_k^-)(\boldsymbol{\chi}_{i,k|k-1}^* - \hat{\mathbf{x}}_k^-)^T + \mathbf{Q}$ 
33:    end function
34:    function Measurement and parameter Update( $\boldsymbol{\chi}_{k|k-1}^*, \mathbf{R}$ )
35:      $\Upsilon_{k|k-1}^* \leftarrow g(\boldsymbol{\chi}_{k|k-1}^*)$ 
36:      $\hat{\mathbf{y}}_k^- \leftarrow \sum_{i=0}^{2L} W_i^{(m)} \Upsilon_{i,k|k-1}^*$ 
37:      $\mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} \leftarrow \sum_{i=0}^{2L} W_i^{(c)} (\Upsilon_{i,k|k-1}^* - \hat{\mathbf{y}}_k^-)(\Upsilon_{i,k|k-1}^* - \hat{\mathbf{y}}_k^-)^T + \mathbf{R}$ 
38:      $\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} \leftarrow \sum_{i=0}^{2L} W_i^{(c)} (\boldsymbol{\chi}_{i,k|k-1}^* - \hat{\mathbf{x}}_k^-)(\Upsilon_{i,k|k-1}^* - \hat{\mathbf{y}}_k^-)^T$ 
39:      $\mathbf{K}_k \leftarrow \mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} \mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^{-1}$ 
40:      $\hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$ 
41:      $\mathbf{P}_k \leftarrow \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} \mathbf{K}_k^T$ 
42:      $\boldsymbol{\chi}_{\theta_i} = \hat{\boldsymbol{\theta}}_{k-1} - \xi^T (\mathbf{y}_k - \Upsilon_{i,k|k-1}^*)$ 
43:      $\hat{\boldsymbol{\theta}}_k = \frac{1}{2L+1} \sum_{i=0}^{2L} \boldsymbol{\chi}_{\theta_i}$ 
44:    end function
45:   end for
46: end function

```

A. VAN DER POL OSCILLATOR

First example is the Van der Pol oscillator. This system is a highly nonlinear system and it is used in [24] as an example to

explain state estimation by using UKF. Van der pol equation is expressed as

$$\dot{x}_1 = x_2, \quad (68a)$$

$$\dot{x}_2 = -x_1 + \mu(1 - x_1^2)x_2. \quad (68b)$$

A phase portrait analysis is carried out for this system in [25]. Hereby, a simulation study to estimate parameter μ with JUKF and MJUKF is presented.

The number of states here is 2 and in JUKF, the length of the state vector becomes 3 by concatenating parameter to the state vector. This means that the number of sigma points will be $2L + 1 = 7$ in JUKF. It is assumed that both states are measured, i.e.

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x. \quad (69)$$

During simulations Gaussian white noise with a 10% signal-to-noise ratio is added to the measurements. Initial state is chosen as

$$x_0 = [1.4 \quad 0], \quad (70)$$

real parameter for the system is

$$\mu = 0.2, \quad (71)$$

and initial state covariance matrix for JUKF is chosen as

$$P_{x_0} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}. \quad (72)$$

Process noise and measurement noise covariances are selected as $Q = 10^{-3} \times I_3$, $R = 10^{-1} \times I_2$, respectively. I_2 and I_3 represent 2×2 and 3×3 identity matrices, respectively. Initial state estimates are selected as

$$\hat{x}_0 = [\hat{x}_1 \quad \hat{x}_2 \quad \hat{\mu}], \quad (73a)$$

$$\hat{x}_0 = [0 \quad 5 \quad 5]. \quad (73b)$$

State and parameter estimation results for JUKF are presented in Fig. 3, 4 and 5, respectively for given initial conditions. It can be seen that initial state and parameter estimate are selected distant from the real values.

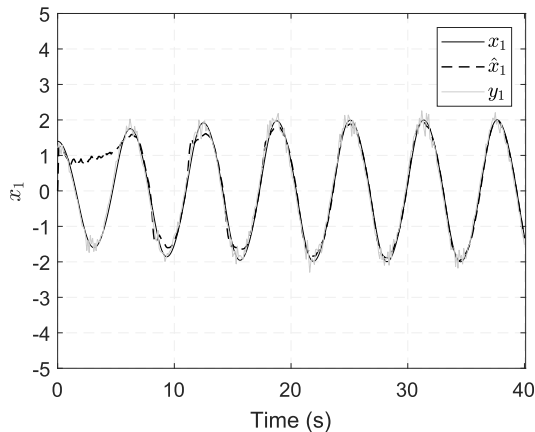


FIGURE 3. First state, state estimate and measurement for Van der Pol oscillator.

In case of MJUKF, since parameter is separated from the state vector, the size of state sigma point matrix propagated

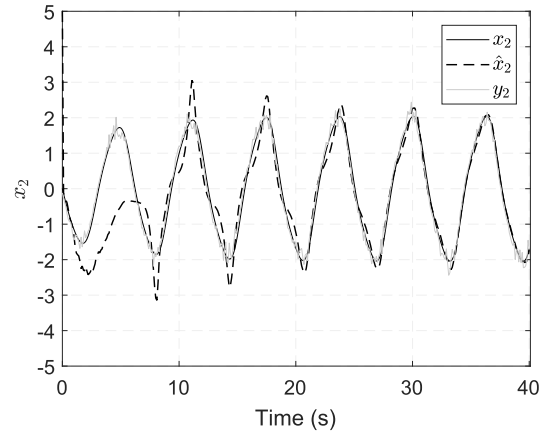


FIGURE 4. Second state, state estimate and measurement for Van der Pol oscillator.

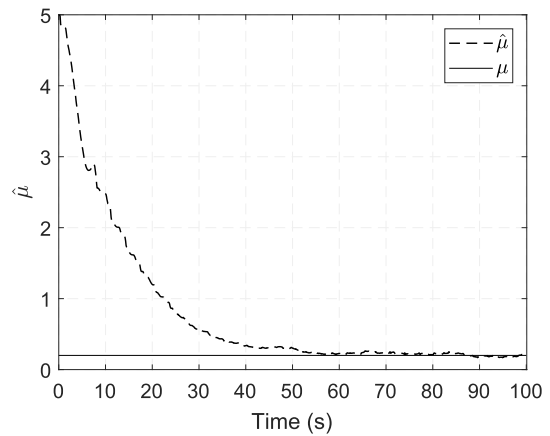


FIGURE 5. Parameter estimate for Van der Pol oscillator with JUKF.

through nonlinear function is 5×2 and the size of parameter sigma point matrix is 5×1 . The determination of parameter sigma point matrix is given in (5) and (6) and it should be noted that no matrix square root operation is carried out. The initial state error covariance is selected with same magnitude in MJUKF as it is in (72)

$$P_{x_0} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, \quad (74)$$

and p_θ for parameter initial error covariance is selected as 0.5 similar to the previous case. In this example, since there is only one parameter, the size of p_θ is 1×1 . Therefore, as given in (5) and (6), initial parameter sigma point matrix is equal to

$$X_\theta = \begin{bmatrix} 4 \\ 4.5 \\ 5 \\ 5.5 \\ 6 \end{bmatrix}. \quad (75)$$

In order to determine the transformation matrix T , first method mentioned in previous section, which is based on obtaining Jacobian matrix, is considered. It should be noted that the nonlinear functions for measurements cannot be

obtained since measurements are equal to the system states and they are given as a differential equation in (68). As stated in previous section, the nonlinear function for system states, in which parameters explicitly appear, can be considered for Jacobian matrix for such case. Even though equations for system states (i.e. measurements) include the state derivatives, it is obvious that Jacobian matrix for these derivatives include information about the measurements. Jacobian matrix for this system can be expressed as

$$\mathbf{J}(\mu) = \begin{bmatrix} \frac{\partial \dot{x}_1}{\partial \mu} \\ \frac{\partial \dot{x}_2}{\partial \mu} \end{bmatrix} = \begin{bmatrix} 0 \\ (1 - x_1^2)x_2 \end{bmatrix}, \quad (76)$$

and it is apparent from the Jacobian matrix that derivative of the second state (i.e. measurement) is influenced by the parameter. This indicates that the parameter certainly influences the second measurement. Thus, $L_\theta \times L_m$ transformation matrix T can be chosen as

$$T = \begin{bmatrix} 0.1 & 0.9 \end{bmatrix}. \quad (77)$$

The sum of elements in the first row of T , which corresponds to the row for μ , is equal to unity and it is compliant with the (67). Jacobian matrix reveals that μ influences second measurement, so t_{12} is selected higher than t_{11} . However, a direct conclusion cannot be drawn about the first measurement since parameter does not explicitly appear in system equations for the first state.

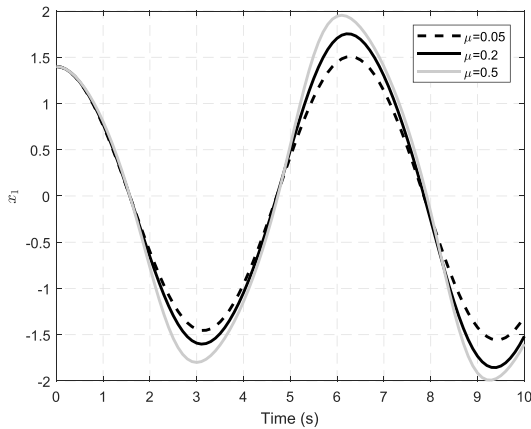


FIGURE 6. Effect of the parameter μ for the first state (i.e. first measurement).

Second method to determine T is to inspect the behavior of the measurements with respect to the parameters by means of simulations. Behavior of the system states (i.e. measurements) with respect to the parameter is illustrated in Fig. 6 and 7. It is clear from both measurements that the magnitudes of measurements increase with respect to increasing μ . Results in these figures seem to contradict with the conclusion drawn in previous paragraphs, but it should be noted that μ does not explicitly appear for the first state in system equations. Therefore, a direct conclusion cannot be

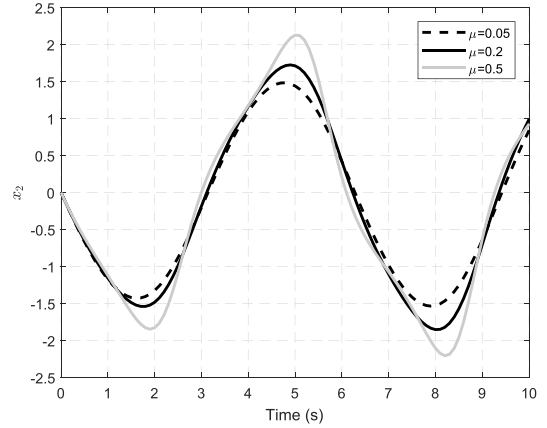


FIGURE 7. Effect of the parameter μ for the second state (i.e. second measurement).

drawn for the first measurement by considering the Jacobian matrix. However, the derivative of the first state is simply equal to the second state ($\dot{x}_1 = x_2$) as given in (68). Thus, a change in the second state affects the first state. T is selected as given in (78) by considering the second method and the elements of T are divided equally since both measurements are influenced by the parameter. Both transformation matrices can be considered for the estimation. Since first measurement also include information about the parameter μ , even a transformation matrix, which t_{11} is higher than t_{12} , can be selected. However, it is suggested in this study that Jacobian matrix is enough to determine transformation matrix if at least there is one measurement influenced by each parameter and it appears explicitly in system equations. T can be expressed in case of determination by simulations

$$T = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}. \quad (78)$$

ξ of this system is determined based on the magnitude of measurements, parameter and the noise present in measurements, and it is selected as $\xi = 0.25$. This parameter can be determined based on simulations or intuitively, and it is critical for the filter stability. If the magnitude of ξ is selected very small, parameter convergence is degraded. Furthermore, if the magnitude of ξ is determined very high, system becomes unstable. State estimation and parameter estimation results for MJUKF, when $T = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$ and $\xi = 0.25$, are given in Fig. 8, 9, and 12, respectively.

In Fig. 10 and 11, the effect of different scaling parameter ξ selection on parameter estimation is presented. When $\xi = 0.4$, system becomes unstable. In this study, ξ is determined based on the system simulations carried out with MJUKF. Moreover, selection of T with two different methods are compared in Fig. 12 and 13. It is obvious that parameter estimates for both matrices converges to real value, but when elements of T are chosen equally as 0.5 convergence of parameter estimation is better since both of the measurements, which are affected by the parameter, are considered. However, ξ is same for both cases and equals to $\xi = 0.25$;

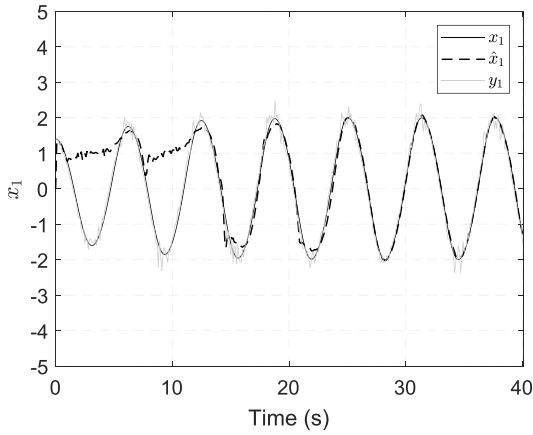


FIGURE 8. First state, state estimate and measurement for Van der Pol oscillator with MJUKF, $\xi = 0.25$.

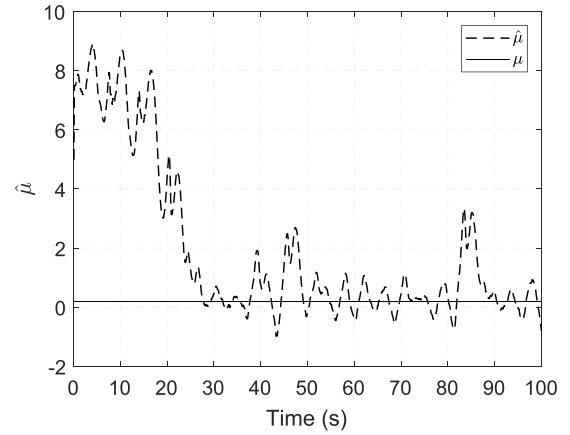


FIGURE 11. Parameter estimate for Van der Pol oscillator with MJUKF for $\xi = 0.35$.

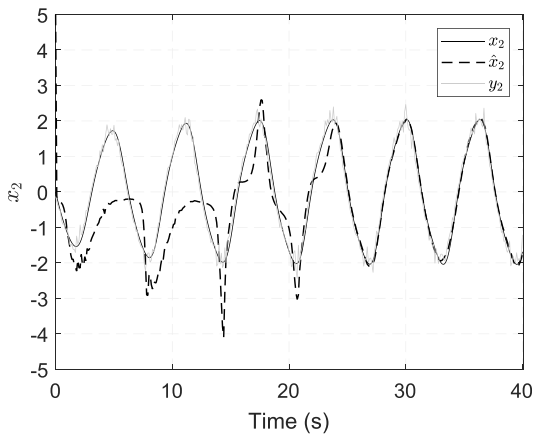


FIGURE 9. Second state, state estimate and measurement for Van der Pol oscillator with MJUKF, $\xi = 0.25$.

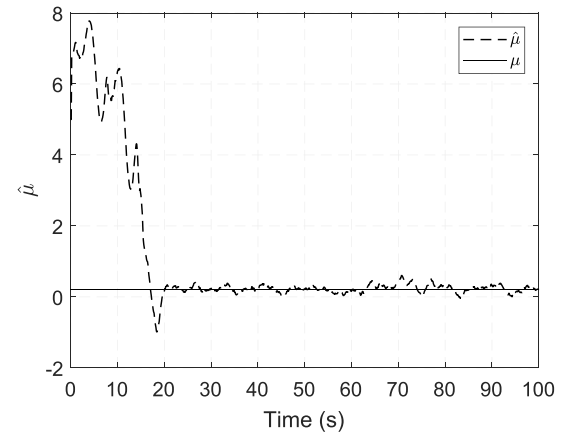


FIGURE 12. Parameter estimate for Van der Pol oscillator with MJUKF for $T = [0.5 \ 0.5]$ and $\xi = 0.25$.

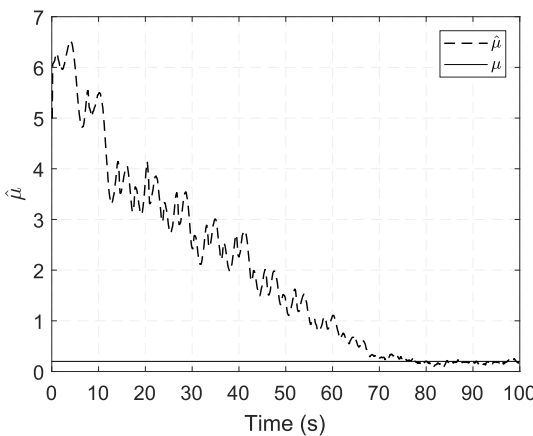


FIGURE 10. Parameter estimate for Van der Pol oscillator with MJUKF for $\xi = 0.15$.

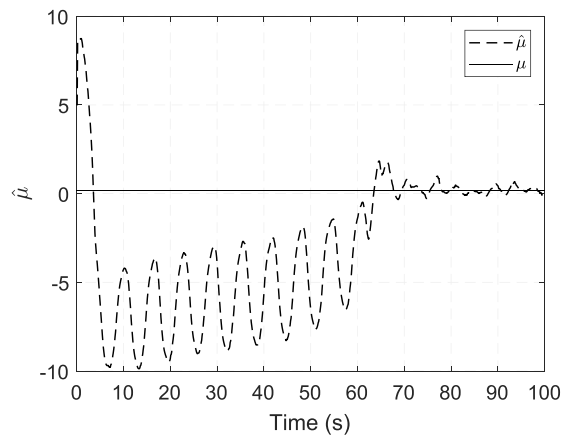


FIGURE 13. Parameter estimate for Van der Pol oscillator with MJUKF for $T = [0.1 \ 0.9]$ and $\xi = 0.25$.

convergence rate for the case $T = [0.1 \ 0.9]$ can be further improved by selecting a ξ with higher magnitude.

In order to compare computational complexity of two algorithms, the stopwatch timer in MATLAB software is used.

Same code is run for JUKF and MJUKF on a MacBook Pro Late 2011 laptop with a 2.8 GHz dual-core Intel Core i7 processor and 8 Gb of RAM. Previously indicated modifications are included in the code for MJUKF. A 100 s simulation

with a 0.1 s time step for Van der Pol oscillator is repeated 10^3 times for both cases. Average elapsed time for both cases are recorded. For parameter estimation in Van der Pol oscillator, average elapsed time for JUKF is 0.4791 s, whereas for MJUKF, average elapsed time is 0.3830 s. When two algorithms are compared with respect to each other, average elapsed time in this new scheme is approximately 20% lower than JUKF. It should be noted that there is only one parameter to be estimated in this case. After 10^3 simulations, mean squared error (MSE) for both cases are illustrated in Fig. 14. It is obvious that parameter estimates converge when a Gaussian white noise with a 10% signal-to-noise ratio is added to the measurements in both cases. However, a comparison with DUKF is not provided since DUKF is unstable when a Gaussian white noise with a 10% signal-to-noise ratio is added to the measurements.

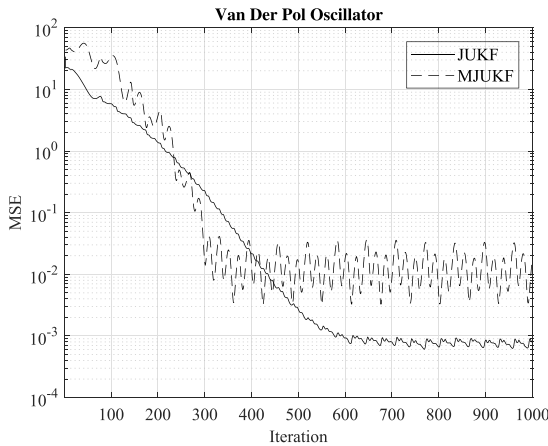


FIGURE 14. MSE curves for Van der Pol oscillator when Gaussian white noise with a 10% signal-to-noise ratio is added to measurements, after 10^3 simulations.

In order to make a comparison between MJUKF, JUKF and DUKF, Gaussian white noise with a 0.1% signal-to-noise ratio is added to the measurements only. Additionally, all parameters are same as the first simulation case, but measurement noise covariance is selected as $\mathbf{R} = 10^{-3} \times \mathbf{I}_2$ for all methods different from the first simulation case with 10% noise level, since DUKF is unstable for the selection of measurement noise covariance as $\mathbf{R} = 10^{-1} \times \mathbf{I}_2$. Noise level in the measurements is changed, so new scaling parameter should be chosen for MJUKF, and it is determined as $\xi = -0.65$. Parameter estimates for MJUKF, JUKF and DUKF are illustrated in Fig. 15, 16 and 17, respectively. Simulations are repeated 10^3 times for 100 s simulation in all approaches, and average elapsed time and MSE are recorded. Results are provided in Table 1.

In order to show robustness of the new scheme to the estimated initial conditions, they are changed to

$$\hat{\mathbf{x}}_0 = [-1 \ 3], \quad (79a)$$

$$\hat{\mu}_0 = -2, \quad (79b)$$

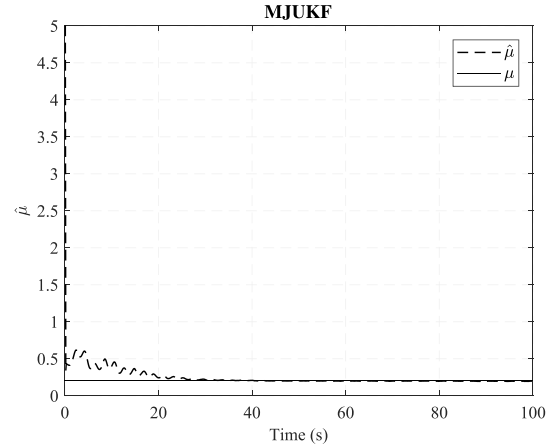


FIGURE 15. Parameter estimate for Van der Pol oscillator with MJUKF (Gaussian white noise with a 0.1% signal-to-noise ratio is added to the measurements).

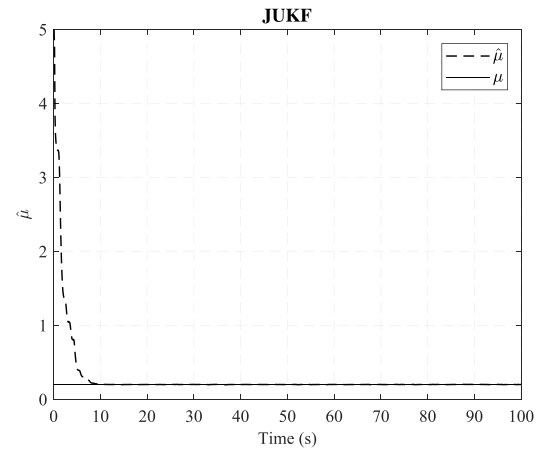


FIGURE 16. Parameter estimate for Van der Pol oscillator with JUKF (Gaussian white noise with a 0.1% signal-to-noise ratio is added to the measurements).

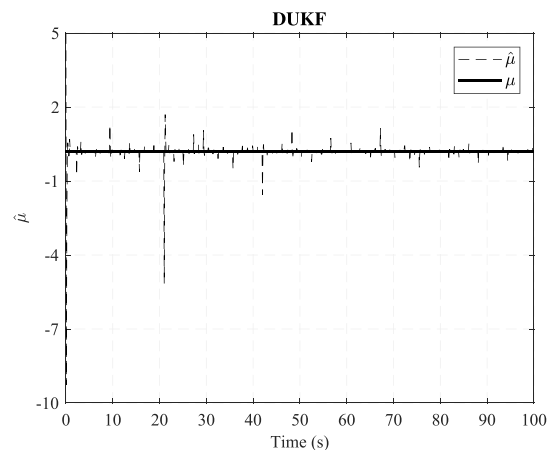


FIGURE 17. Parameter estimate for Van der Pol oscillator with DUKF (Gaussian white noise with a 0.1% signal-to-noise ratio is added to the measurements).

and results for this new initial conditions can be found in Fig. 18, 19 and 20. Filter parameters for MJUKF are $T = [0.5 \ 0.5]$ and $\xi = 0.25$.

TABLE 1. Average elapsed time and MSE for MJUKF, JUKF, DUKF for 10^3 repetitions of simulations for Gaussian white noise with a 0.1% signal-to-noise ratio is added to the measurements.

	MJUKF	JUKF	DUKF
Average elapsed time in seconds	0.4049	0.4721	0.8696
MSE for parameter estimation	0.0553	0.1569	0.2278

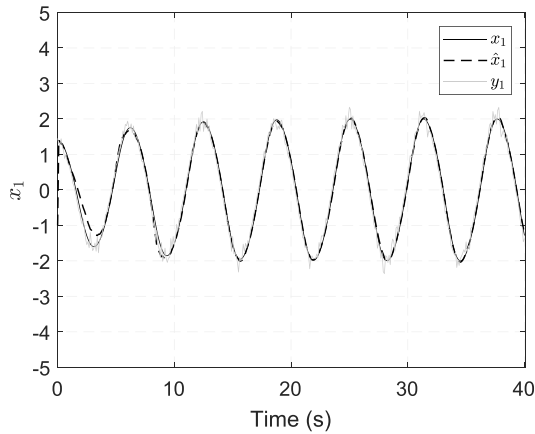


FIGURE 18. First state estimate for Van der Pol oscillator with MJUKF for different initial condition estimates.

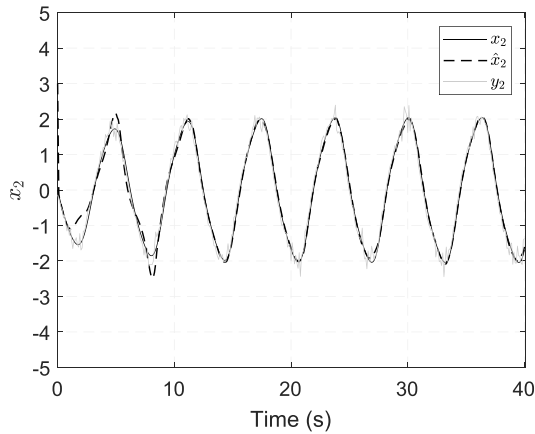


FIGURE 19. Second state estimate for Van der Pol oscillator with MJUKF for different initial condition estimates.

Next, the robustness of the proposed algorithm for parameter changes are tested. At $t = 40$ s, real parameter value changes from 0.2 to 2. Parameter estimation result of MJUKF is given in Fig. 21 for the first estimation case (i.e. $\hat{\mathbf{x}}_0 = [0 \ 5]$, $\hat{\mu}_0 = 5$).

B. LORENZ 63 SYSTEM

Next, MJUKF is applied for the parameter estimation in Lorenz 63 system, given in [23] and [26]. It includes 3 states and 3 parameters. As stated in [23], this system has strong nonlinearities and chaotic behavior. It is expressed with the differential equations

$$\dot{x} = \sigma(y - x), \quad (80a)$$

$$\dot{y} = x(\rho - z) - y, \quad (80b)$$

$$\dot{z} = xy - \beta z. \quad (80c)$$

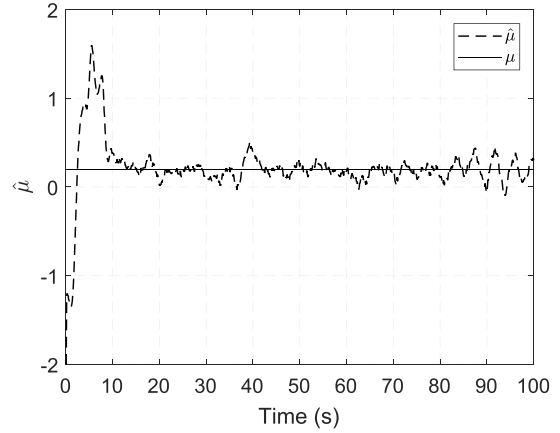


FIGURE 20. Parameter estimate for Van der Pol oscillator with MJUKF for different initial condition estimates.

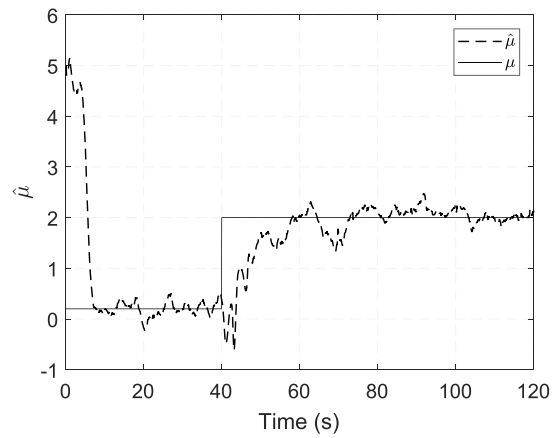


FIGURE 21. Parameter estimate for Van der Pol oscillator with MJUKF, at $t = 40$ s μ becomes 2.

There are 3 states and 3 parameters and by concatenating state and parameter vector, the size of the state vector becomes 6 for JUKF. The number of sigma points is $2L + 1 = 13$ for JUKF. Similar to the first example, it is assumed that all states are measured, and measurement vector is given as

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}. \quad (81)$$

During simulations, Gaussian white noise with a 10% signal-to-noise ratio is added to the measurements only. Initial state and parameter vector are chosen same as given in [23]

$$\mathbf{x}_0 = [x \ y \ z] = [0.9 \ 1 \ 1.1], \quad (82)$$

real parameters for the system are

$$\theta = [\sigma \ \rho \ \beta] = [10 \ 28 \ 8/3]. \quad (83)$$

initial state covariance matrix is chosen for JUKF as

$$\mathbf{P}_{x_0} = \begin{bmatrix} 10^{-3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}, \quad (84)$$

process noise and measurement noise covariances are selected as $\mathbf{Q} = 10^{-3} \times I_3$, $\mathbf{R} = 10^{-1} \times I_3$, respectively. Initial state estimates are selected as

$$\hat{\mathbf{x}}_0 = [\hat{x} \quad \hat{y} \quad \hat{z} \quad \hat{\sigma} \quad \hat{\rho} \quad \hat{\beta}], \quad (85a)$$

$$\hat{\mathbf{x}}_0 = [1.5 \quad 1.5 \quad 1.5 \quad 5 \quad 21 \quad 1/3]. \quad (85b)$$

State estimation results for similar simulation scenario can be found in [23]. Since the aim in this study is to estimate parameters, only parameter estimation results are presented, and they can be seen in Fig. 22, 23 and 24 for JUKF.

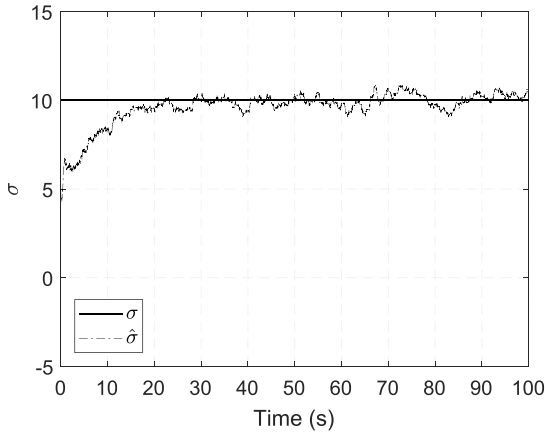


FIGURE 22. Real value of σ and its estimate-JUKF.

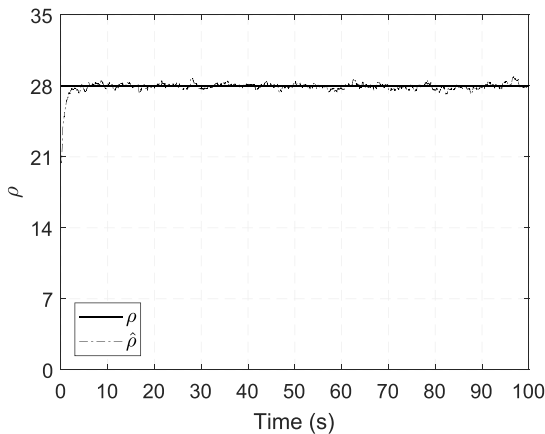


FIGURE 23. Real value of ρ and its estimate-JUKF.

For the same initial state and parameter estimates, now the results for MJUKF are provided in Fig. 25, 26 and 27. In this case, since parameter vector is separated from state vector, the number of sigma points propagated through nonlinear

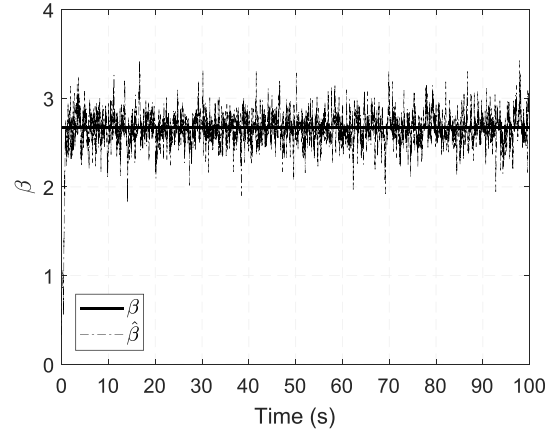


FIGURE 24. Real value of β and its estimate-JUKF.

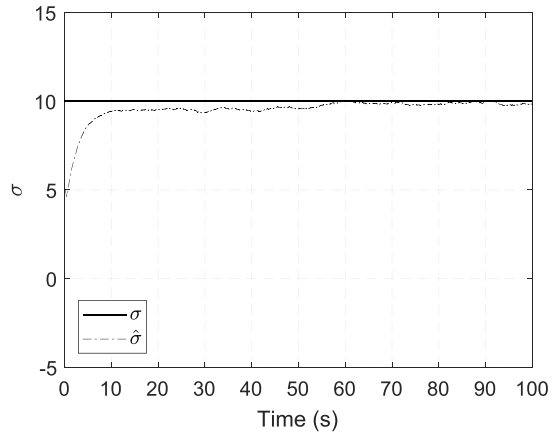


FIGURE 25. Real value of σ and its estimate-MJUKF.

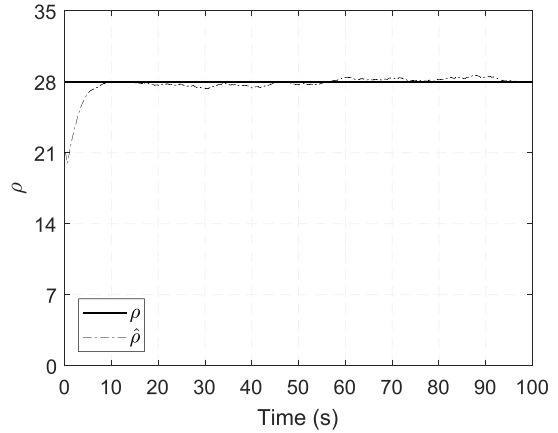


FIGURE 26. Real value of ρ and its estimate-MJUKF.

function is 7×3 and the size of parameter sigma point matrix is 7×3 . The determination of initial parameter sigma points is given in (5) and (6). It should be noted that no matrix square root operation is necessary. The initial state error covariance is selected with same magnitude as it is in (84)

$$\mathbf{P}_{x_0} = \begin{bmatrix} 10^{-3} & 0 & 0 \\ 0 & 10^{-3} & 0 \\ 0 & 0 & 10^{-3} \end{bmatrix}, \quad (86)$$

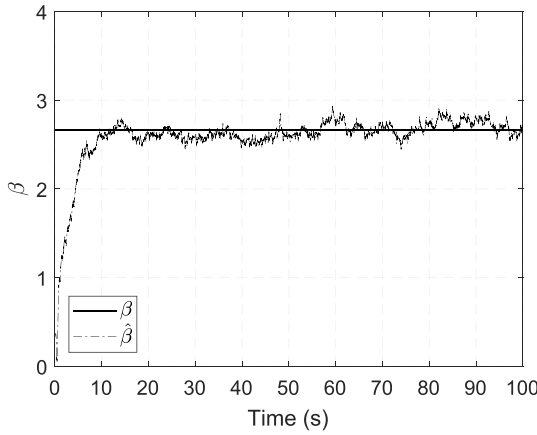


FIGURE 27. Real value of β and its estimate–MJUKF.

and elements of \mathbf{p}_θ for parameter initial error covariance is selected as 0.5 similar to the previous case. In this example, since there are three parameters, the size of \mathbf{p}_θ is 1×3 and it is equal to $\mathbf{p}_\theta = [0.5 \ 0.5 \ 0.5]$. Therefore, as given in (5) and (6), initial parameter sigma point matrix is equal to

$$\chi_\theta = \begin{bmatrix} 3.5 & 19.5 & -1.1667 \\ 4 & 20 & -0.6667 \\ 4.5 & 20.5 & -0.1667 \\ 5 & 21 & 0.3333 \\ 5.5 & 21.5 & 0.8333 \\ 6 & 22 & 1.3333 \\ 6.5 & 22.5 & 1.8333 \end{bmatrix}. \quad (87)$$

It should be noted that $\hat{\theta}_0 = [\hat{\sigma}_0 \ \hat{\rho}_0 \ \hat{\beta}_0] = [5 \ 21 \ 1/3]$. Transformation matrix has size of 3×3 (i.e. $L_m \times L_\theta$) and it is determined based on the Jacobian matrix. Same situation is valid for Lorenz 63 system that measurements are equal to states, and they are given as a differential equations. Therefore, nonlinear functions for system states (i.e. measurements) are used to obtain Jacobian matrix and it is expressed as

$$\mathbf{J}(\sigma, \rho, \beta) = \begin{bmatrix} \frac{\partial \dot{x}}{\partial \sigma} & \frac{\partial \dot{x}}{\partial \rho} & \frac{\partial \dot{x}}{\partial \beta} \\ \frac{\partial \dot{y}}{\partial \sigma} & \frac{\partial \dot{y}}{\partial \rho} & \frac{\partial \dot{y}}{\partial \beta} \\ \frac{\partial \dot{z}}{\partial \sigma} & \frac{\partial \dot{z}}{\partial \rho} & \frac{\partial \dot{z}}{\partial \beta} \end{bmatrix} = \begin{bmatrix} y-x & 0 & 0 \\ 0 & x & 0 \\ 0 & 0 & -z \end{bmatrix}. \quad (88)$$

There is at least one measurement for each column (i.e. parameter), so Jacobian matrix can be used to determine T . It is apparent from the (88) that each state x, y, z is influenced by each parameter σ, ρ, β , respectively. Since each estimated parameter affects directly corresponding state (e.g. σ affects parameter x), weights of the other states are chosen low, intuitively. Thus, elements in T corresponding to estimated parameters are selected higher than others.

T is given as

$$T = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}. \quad (89)$$

Obviously, transformation matrix in (89) satisfies the rule given in (67), and the convergence of parameter estimates are presented in Fig. 25, 26, 27 with a $\xi = 0.006$. Second method to obtain T based on simulations is not considered for this system since there is at least one measurement influenced by each parameter in Jacobian matrix, and it is enough for the estimation purpose. Corresponding elements of the transformation matrix can also be selected as unity which implies that only the measurement affected by the corresponding parameter is considered. This selection also satisfies the conditions presented in (64) and (67), so that still the parameter update rule proposed in (20) provides MAP estimates for each parameter sigma point. In this case, T is an identity matrix and it makes the selection of filter parameters of MJUKF simple. Thus, only the selection of ξ is required in this case and it is selected same as the previous case $\xi = 0.006$. By considering the same initial conditions and the following transformation matrix

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (90)$$

the parameter estimation results can be seen in Fig. 28, 29, 30.

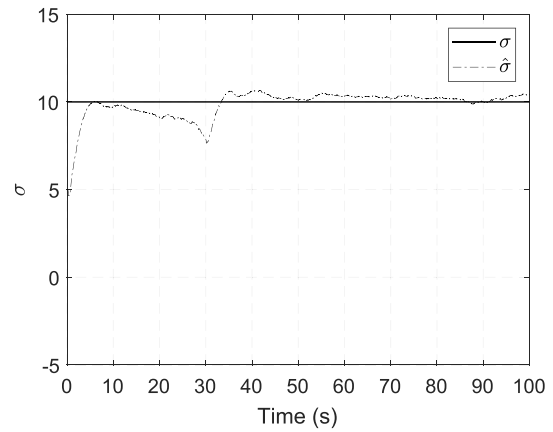


FIGURE 28. Real value of σ and its estimate–MJUKF when T is selected as identity matrix given in (90).

The scaling parameter ξ is determined based on the magnitude of states and parameters and selected as $\xi = 0.006$ by carrying out several simulations with MJUKF. Determination of this parameter is intuitive and again it is critical for the system stability. Similarly, in order to show robustness of MJUKF to different initial states and parameter estimates, state and parameter estimates are selected as

$$\hat{\mathbf{x}}_0 = [0.5 \ 0.5 \ 0.5], \quad (91a)$$

$$\hat{\theta}_0 = [13 \ 33 \ 10/3], \quad (91b)$$

and results can be seen in Fig. 31, 32 and 33.

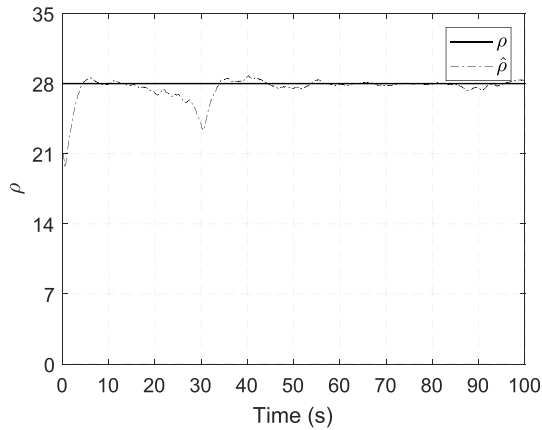


FIGURE 29. Real value of ρ and its estimate-MJUKF when T is selected as identity matrix given in (90).

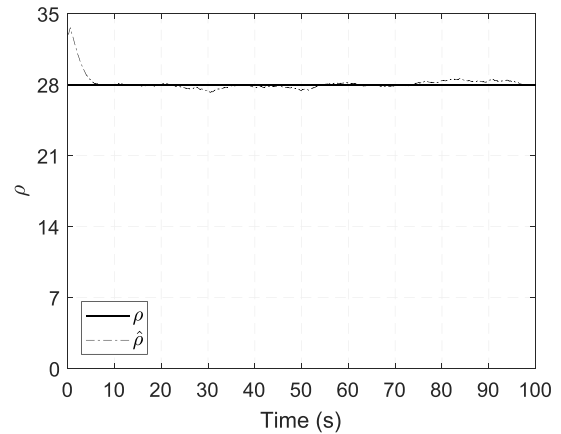


FIGURE 32. Real value of ρ and its estimate for Lorenz system with MJUKF for different state and parameter estimates.

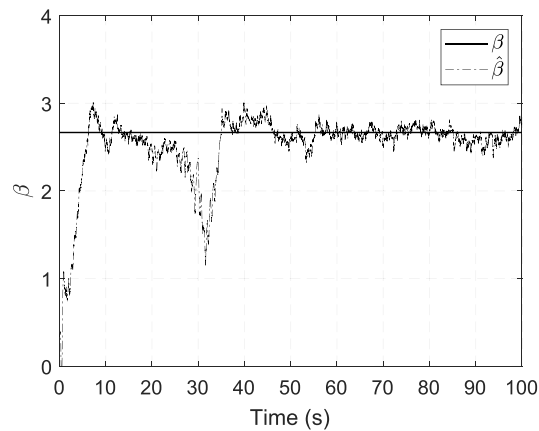


FIGURE 30. Real value of β and its estimate-MJUKF when T is selected as identity matrix given in (90).

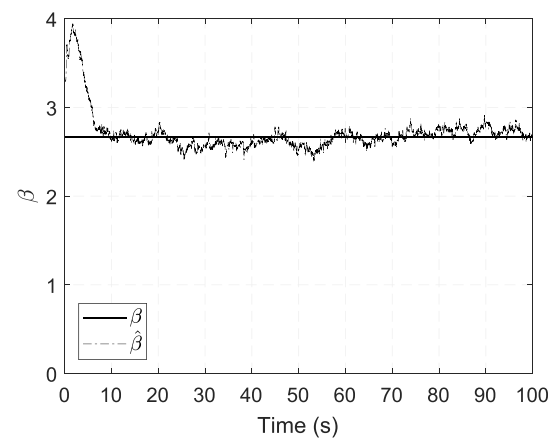


FIGURE 33. Real value of β and its estimate for Lorenz system with MJUKF for different state and parameter estimates.

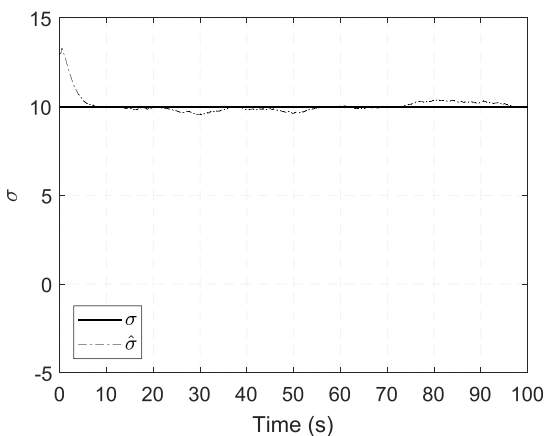


FIGURE 31. Real value of σ and its estimate for Lorenz system with MJUKF for different state and parameter estimates.

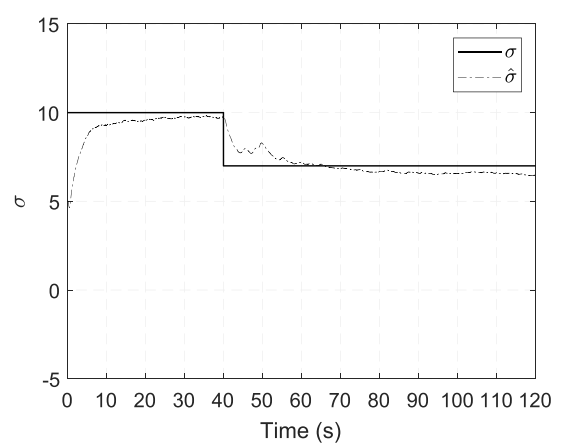


FIGURE 34. Real value of σ and its estimate for Lorenz system with MJUKF, at $t = 40$ s real parameters change.

The same scenario provided in [23] is simulated. Initial state estimates are again $\hat{\mathbf{x}}_0 = [1.5 \ 1.5 \ 1.5]$ and initial parameter estimates are $\hat{\theta}_0 = [\hat{\sigma}_0 \ \hat{\rho}_0 \ \hat{\beta}_0] = [5 \ 21 \ 1/3]$. At $t = 40$ s, real parameter values change

from $\theta = [10 \ 28 \ 8/3]$ to $\theta = [7 \ 21 \ 1]$. Estimation results for this case are presented in Fig. 34, 35 and 36.

Similar to the Van der Pol oscillator case, simulations are repeated 10^3 times with same computer. Hereby, a 100 s

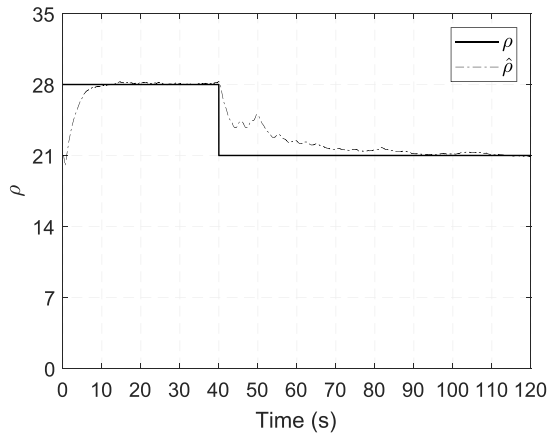


FIGURE 35. Real value of ρ and its estimate for Lorenz system with MJUKF, at $t = 40$ s real parameters change.

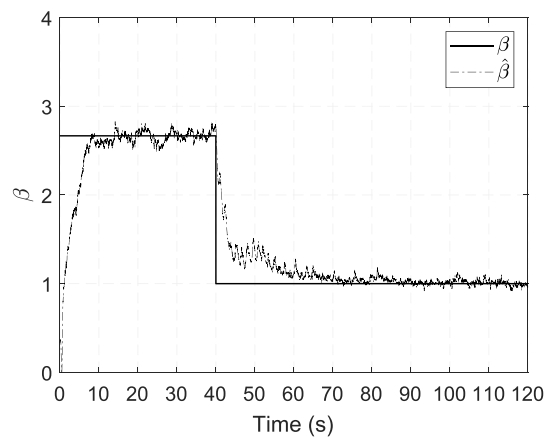


FIGURE 36. Real value of β and its estimate for Lorenz system with MJUKF, at $t = 40$ s real parameters change.

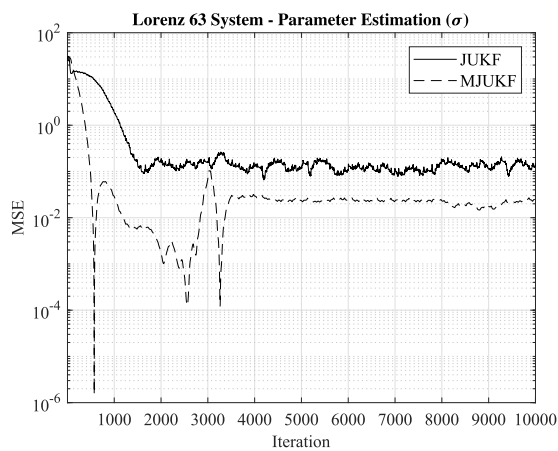


FIGURE 37. MSEs of $\hat{\sigma}$ for MJUKF and JUKF for Lorenz 63 system.

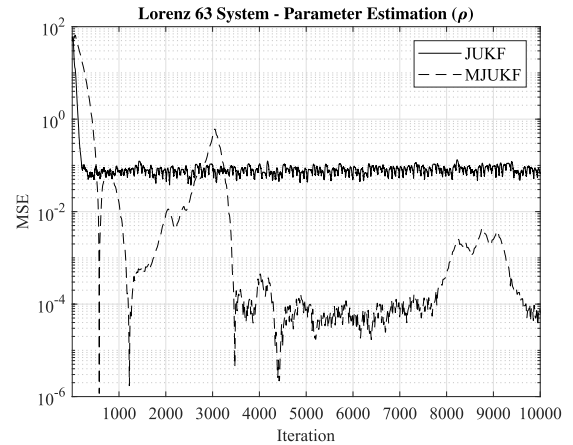


FIGURE 38. MSEs of $\hat{\rho}$ for MJUKF and JUKF for Lorenz 63 system.

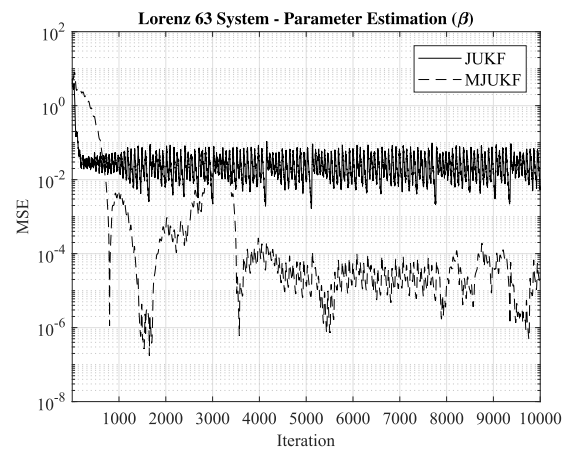


FIGURE 39. MSEs of $\hat{\beta}$ for MJUKF and JUKF for Lorenz 63 system.

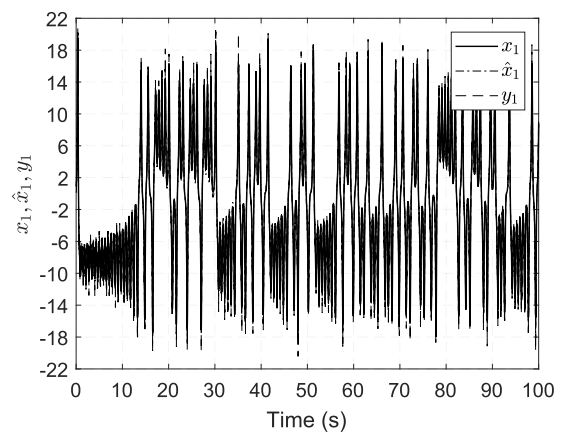


FIGURE 40. First state, state estimate with MJUKF and measurement for initial state and parameter estimate given in (92) and (93), respectively.

simulation with a 0.01 s time step is considered. Average elapsed time during repetitions is recorded. For parameter estimation in Lorenz system, average elapsed time for JUKF is 9.1531 s, whereas for MJUKF, average elapsed time is 5.8625 s. When two algorithms are compared with respect

to each other, average elapsed time in this new scheme is approximately 36% lower than JUKF. MSEs for both cases are illustrated in Fig. 37, 38, 39 for 10^3 simulations. It is obvious that for both cases parameter estimates converge when Gaussian white noise with a 10% signal-to-noise ratio

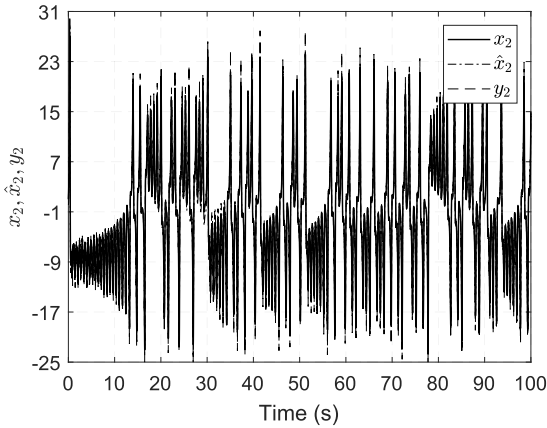


FIGURE 41. Second state, state estimate with MJUKF and measurement for initial state and parameter estimate given in (92) and (93), respectively.

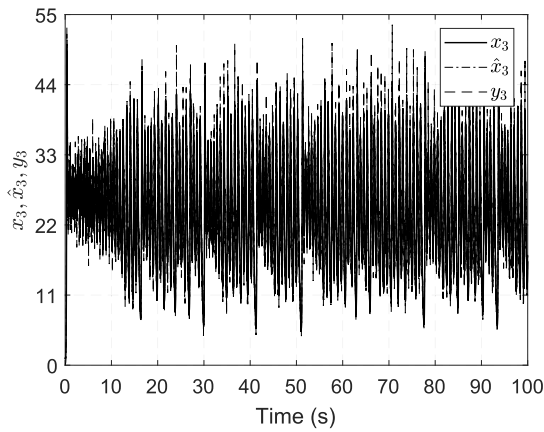


FIGURE 42. Third state, state estimate with MJUKF and measurement for initial state and parameter estimate given in (92) and (93), respectively.

is added to the measurements. However, a comparison with DUKF is not given for Lorenz 63 system since DUKF is expected to provide worse results [5] in terms of accuracy and computational complexity, and such comparison is given for Van Der Pol oscillator.

Lastly, two different scenarios are presented to reveal the consistency of MJUKF. Therefore, in the first case small parametric uncertainty with large state uncertainty and in the second case, small state uncertainty with large parametric uncertainty is considered for the Lorenz 63 system. In the first simulation scenario, $\xi = 0.006$ and T is selected as given in (89). The values of initial states and real parameters are presented in (82) and (83). The initial state estimate is selected far from the real values whereas parameter estimates are selected close to the real values and they can be expressed as

$$\hat{\mathbf{x}}_0 = [0.1 \quad 0.1 \quad 0.1], \quad (92)$$

$$\hat{\boldsymbol{\theta}}_0 = [\hat{\sigma}_0 \quad \hat{\rho}_0 \quad \hat{\beta}_0] = [11 \quad 29 \quad 5/3]. \quad (93)$$

State estimation results are given in Fig. 40, 41, 42. Additionally, state estimation results for first 5 seconds is

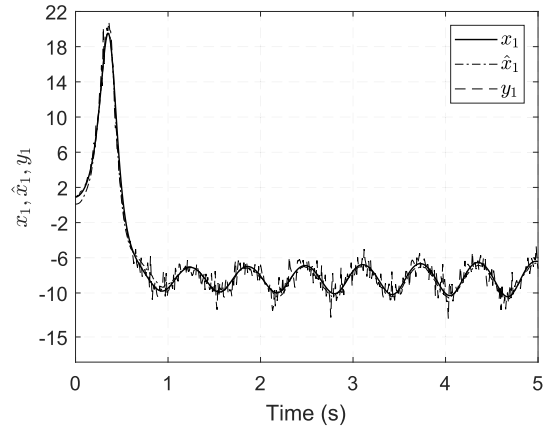


FIGURE 43. First state, state estimate with MJUKF and measurement (5 sec. simulation) for initial state and parameter estimate given in (92) and (93), respectively.

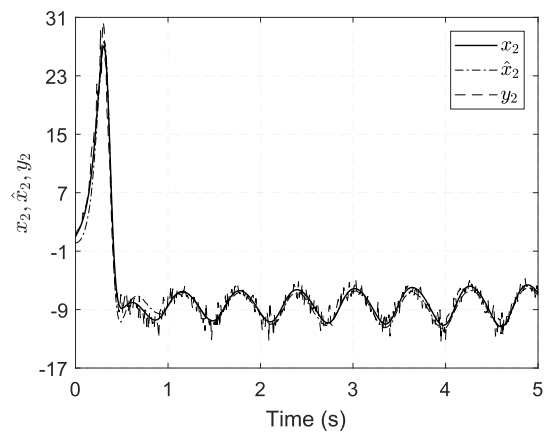


FIGURE 44. Second state, state estimate with MJUKF and measurement (5 sec. simulation) for initial state and parameter estimate given in (92) and (93), respectively.

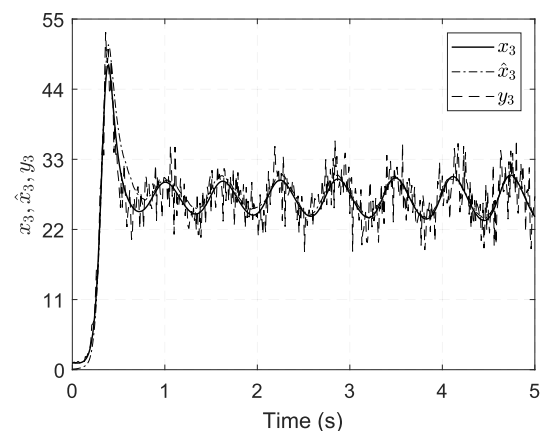


FIGURE 45. Third state, state estimate with MJUKF and measurement (5 sec. simulation) for initial state and parameter estimate given in (92) and (93), respectively.

presented in Fig. 43, 44, 45 in order to clearly inspect the results. In this simulation scenario, parameter estimates are provided in Fig. 46, 47, 48. It is apparent from the simulations

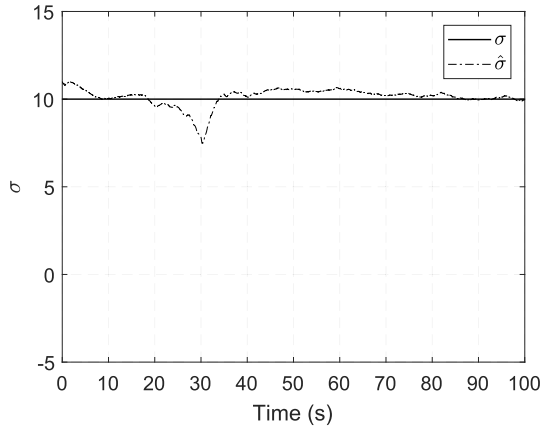


FIGURE 46. Real value of σ and its estimate with MJUKF for initial state and parameter estimate given in (92) and (93), respectively.

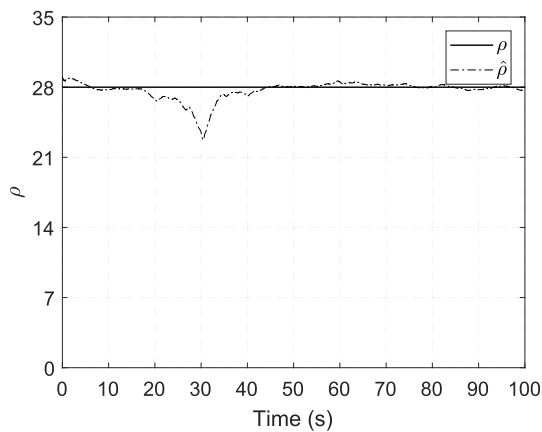


FIGURE 47. Real value of ρ and its estimate with MJUKF for initial state and parameter estimate given in (92) and (93), respectively.

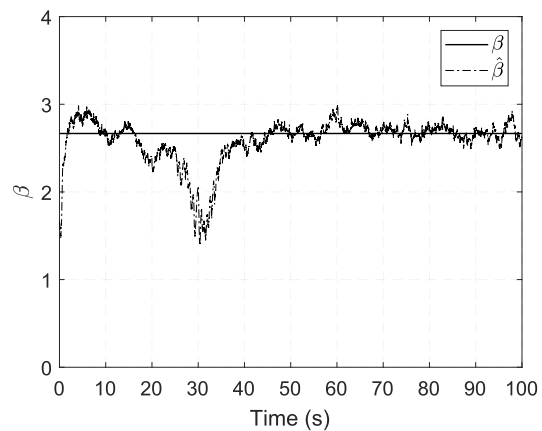


FIGURE 48. Real value of β and its estimate with MJUKF for initial state and parameter estimate given in (92) and (93), respectively.

that the filter is consistent when initial state estimates are selected far from real values whereas parameter estimates are close to the real values.

Different from the first scenario, the initial state and parameter estimates in the second scenario are

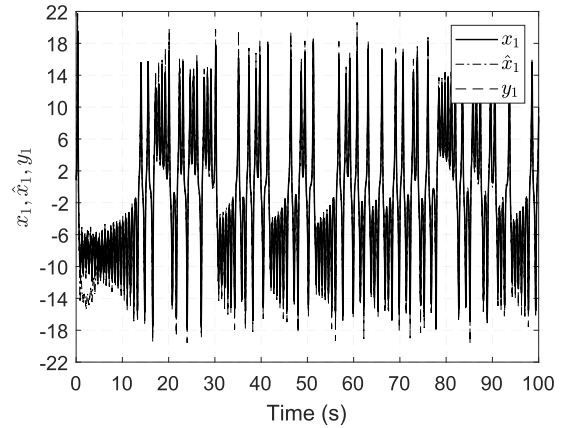


FIGURE 49. First state, state estimate and measurement for initial state and parameter estimate given in (94) and (95), respectively.

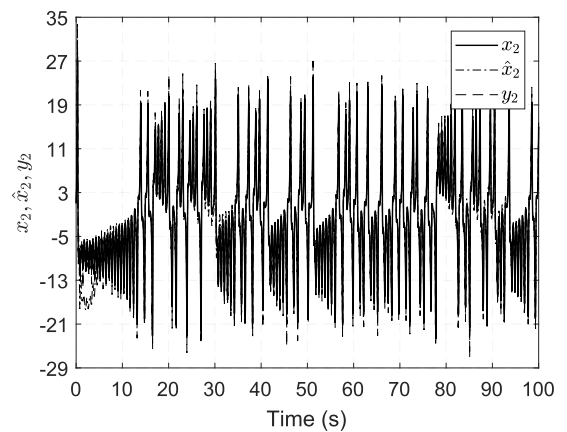


FIGURE 50. Second state, state estimate and measurement for initial state and parameter estimate given in (94) and (95), respectively.

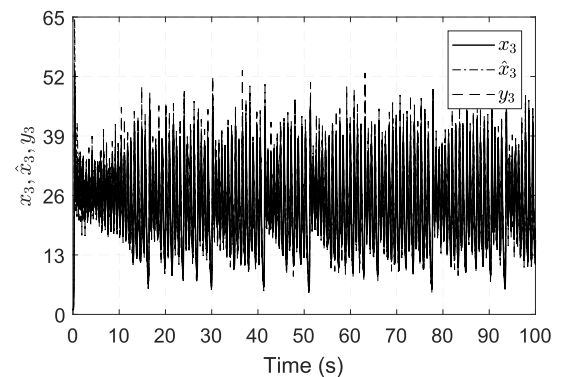


FIGURE 51. Third state, state estimate and measurement for initial state and parameter estimate given in (94) and (95), respectively.

selected as

$$\hat{\mathbf{x}}_0 = [0.8 \quad 1.1 \quad 1], \quad (94)$$

$$\hat{\boldsymbol{\theta}}_0 = [\hat{\sigma}_0 \quad \hat{\rho}_0 \quad \hat{\beta}_0] = [20 \quad 40 \quad 1/30]. \quad (95)$$

On the other hand, except initial state and parameter estimates, all other parameters are same as the first scenario.

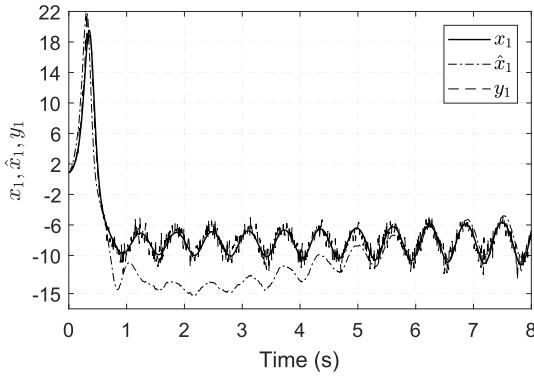


FIGURE 52. First state, state estimate with MJUKF and measurement (8 sec. simulation) for initial state and parameter estimate given in (94) and (95), respectively.

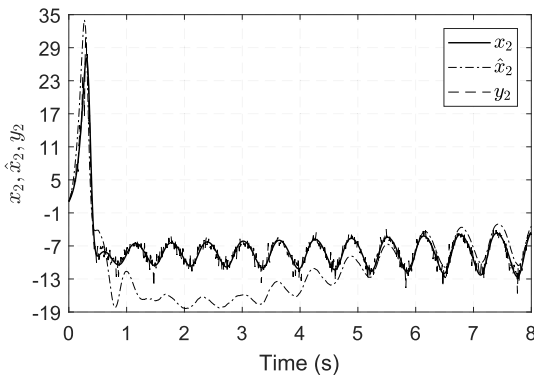


FIGURE 53. Second state, state estimate with MJUKF and measurement (8 sec. simulation) for initial state and parameter estimate given in (94) and (95), respectively.

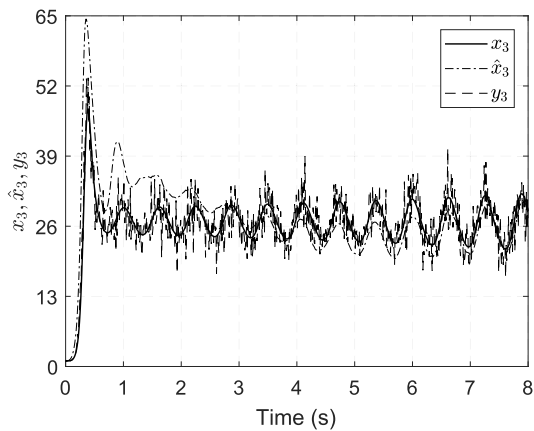


FIGURE 54. Third state, state estimate with MJUKF and measurement (8 sec. simulation) for initial state and parameter estimate given in (94) and (95), respectively.

In this case, initial state estimates are selected close to the real values, whereas initial parameter estimates are selected far from the real values. Similar to the first scenario, state estimation results are presented in Fig. 49, 50, 51. State estimation results for first 8 seconds is given in Fig. 52, 53, 54 in order to clearly inspect the results, and parameter estimates

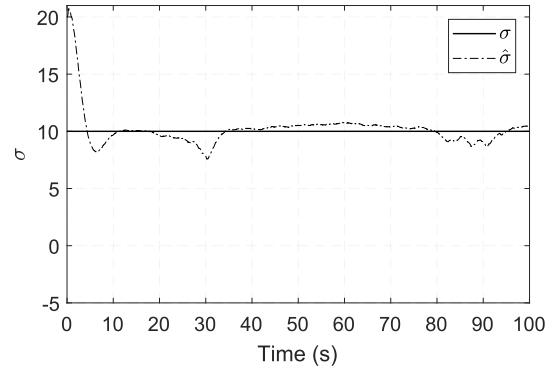


FIGURE 55. Real value of σ and its estimate with MJUKF for initial state and parameter estimate given in (94) and (95), respectively.

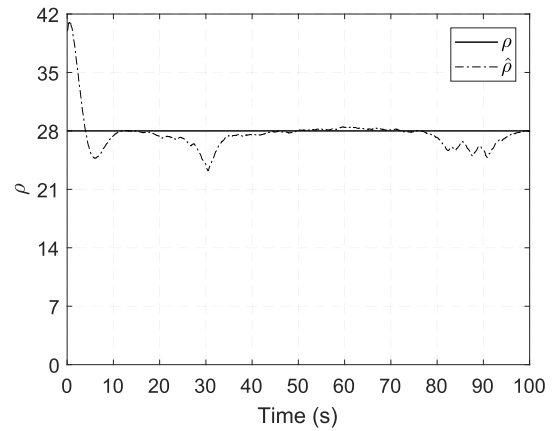


FIGURE 56. Real value of ρ and its estimate with MJUKF for initial state and parameter estimate given in (94) and (95), respectively.

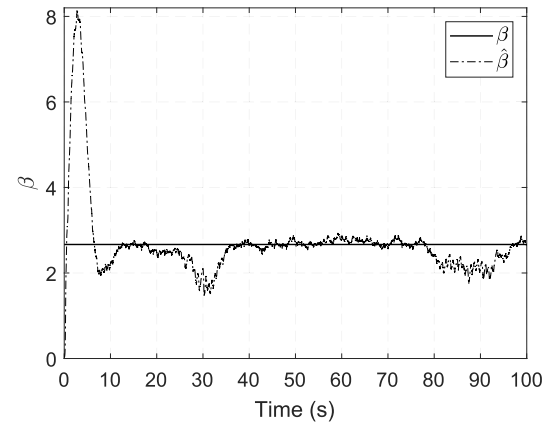


FIGURE 57. Real value of β and its estimate with MJUKF for initial state and parameter estimate given in (94) and (95), respectively.

are provided in Fig. 55, 56, 57. It can be seen from these two simulation scenarios that MJUKF is consistent and provides promising results.

IV. DISCUSSION AND CONCLUSION

It is shown that the modification proposed here provides significant reduction in computational complexity of the JUKF

for parameter estimation. Approximately, 20% reduction for average elapsed time is obtained when MJUKF is applied for Van Der Pol oscillator. When MJUKF is applied to Lorenz 63 system, which includes three states and three parameters, 36% reduction for average elapsed time is obtained. Firstly, Gaussian white noise with a 10% signal-to-noise ratio is added to measurements of Van der Pol oscillator and the results are provided for JUKF and MJUKF. It can be seen from Fig. 14 that parameter estimate converges to real parameter value while providing 20% computational complexity reduction with respect to JUKF. However, parameter estimation result for DUKF, when Gaussian white noise with a 10% signal-to-noise ratio is added to measurements, could not be presented since DUKF is unstable for such noise level. Therefore, Gaussian white noise with a 0.1% signal-to-noise ratio added to measurements and results are presented in Table 1. As previously reported in the literature [5], [21], [27], since there is no cross covariance calculation for states and parameters, DUKF is expected to provide worse results than the JUKF. Nevertheless, a comparison of MJUKF, JUKF and DUKF for Van Der Pol oscillator is provided and results in Table 1 reveal that when noise in the system is low, MJUKF surpasses JUKF and DUKF in terms of accuracy and computational complexity. When noise level is high, both MJUKF and JUKF perform well for parameter estimation, and MJUKF surpasses JUKF in terms of computational complexity while maintaining convergence of parameter estimate as seen in Fig. 14. However, in Fig. 14, it is apparent that JUKF outperform MJUKF in terms of MSE. The main reason behind this fact is that a cross covariance ($\mathbf{P}_{x_k \theta_k}$) between states and parameters is calculated in JUKF. Therefore, JUKF outperforms MJUKF when the noise levels for measurements are high depending on the system. Nevertheless, for low noise levels, MJUKF with a proper selection of T and ξ provides better results in terms of MSE compared to JUKF since a cross covariance matrix ($\mathbf{P}_{x_k \theta_k}$) is not calculated in MJUKF, and calculation of a cross covariance matrix delays the parameter convergence for low noise level in JUKF as shown in Fig. 16. It is provided in Fig. 37, 38, 39 for Lorenz 63 system that MJUKF with a proper selection of T and ξ outperform JUKF for 10% noise level case. Nevertheless, it is observed that parameter estimation results are better for JUKF for higher noise levels than 10% for Lorenz 63 system, even though the results are not reported here for clarity. Therefore, significant advantage of MJUKF is obvious in terms of computational complexity, whereas the performance of MJUKF with respect to JUKF in terms of MSE depends on the noise level for the considered system and proper selection of filter parameters T and ξ .

In this modified scheme, there are two critical parameters. First one is the transformation matrix T and the second one is the scaling parameter ξ . Two heuristic methods are proposed here for the selection of the transformation matrix. First one is based on obtaining a Jacobian matrix for measurements (or states) with respect to parameters. Obviously, such Jacobian matrix can be obtained if parameters explicitly appear

in nonlinear system functions and this method can be used when at least one measurement influenced by each parameter appears in Jacobian matrix. Second method is to observe the relationship between measurements and parameters by means of simulations and then weighting the measurement with respect to corresponding parameter in T . Determination of the scaling parameter ξ is similar to the selection of α in standard UKF which controls the spread of sigma points around the state estimate. It is determined based on the magnitudes and noise levels of measurements, and simulations. Depending on the considered problem, limits for this parameter are determined as $10^{-4} \leq |\xi| \leq 1$ similar to the α in standard UKF [5], [23]. However, adaptive and/or optimal selection of transformation matrix T and scaling parameter ξ are not investigated in this study and it is left as further work. Additionally, UKF, JUKF and MJUKF perform well under *a priori* assumptions and one of these assumptions is the selection of process and measurement noise covariances for these filters. However, *a priori* knowledge of such knowledge is not always possible and one of the approaches to solve this problem is to introduce adaptive rules [28]. In this study, the use of such adaptive mechanisms is recommended if *a priori* knowledge is not present to determine process and measurement noise covariances. Such an adaptive rule to determine these noise covariances is proposed in [28]. Similar approach can be applied for MJUKF.

The purpose of this study is to improve the computational complexity of the standard JUKF with a simple modification. It should be noted that this modification is based on a linear transformation. Unlike JUKF, MJUKF is expected to perform well for the nonlinear systems for which a linear transformation between measurements and parameters is obtained. Fortunately, such transformation is possible for majority of the mechanical, electrical, electromechanical systems etc. Generally, in such systems a constant parameter is inspected for fault diagnosis and condition monitoring [16], [17]. Therefore, MJUKF is especially preferable for these systems if computational complexity is the main concern. Besides, it is proven here for such systems that if the noise levels of the measurements are low, MJUKF outperforms JUKF in terms of computational complexity and MSE.

In conclusion, a modification to the JUKF for additive noise case is presented in this study which is based on decoupling parameter vector from state vector and using it separately. This operation reduces the size of state sigma point matrix and significant reduction in computational complexity is obtained. MJUKF is applied and tested on two dynamic systems commonly used in the literature. While reducing the computational complexity, parameter convergence is guaranteed and it is tested for different cases in simulations. This new scheme is especially preferable to JUKF when there are large number of parameters to be estimated.

ACKNOWLEDGMENT

The author would like to thank to Dr. Asım Önder for proof-reading and checking the text.

REFERENCES

- [1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.
- [2] S. F. Schmidt, "Kalman filter: Its recognition and development for aerospace applications," *J. Guid. Control*, vol. 4, no. 1, pp. 4–7, 1981.
- [3] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.
- [4] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proc. Amer. Control Conf.*, vol. 3, Jun. 1995, pp. 1628–1632.
- [5] R. Van Der Merwe, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," Ph.D. dissertation, Dept. Fac. OGI School Sci. Eng., Oregon Health Sci. Univ., Hillsboro, OR, USA, 2004.
- [6] H. M. T. Menegaz, J. Y. Ishihara, G. A. Borges, and A. N. Vargas, "A Systematization of the Unscented Kalman Filter Theory," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2583–2598, Oct. 2015.
- [7] X. Zhu and E. Feng, "Joint estimation in batch culture by using unscented Kalman filter," *Biotechnol. Bioprocess Eng.*, vol. 17, no. 6, pp. 1238–1243, 2012.
- [8] M. Wielitzka, M. Dagen, and T. Ortmaier, "Joint unscented Kalman filter for state and parameter estimation in vehicle dynamics," in *Proc. IEEE Conf. Control Appl. (CCA)*, Sep. 2015, pp. 1945–1950.
- [9] Y. Yuan, G. Fu, and W. Zhang, "Extended and unscented Kalman filters for parameter estimation of a hydrodynamic model of vessel," in *Proc. 35th Chin. Control Conf. (CCC)*, Jul. 2016, pp. 2051–2056.
- [10] M. Cui, W. Liu, H. Liu, and X. Lü, "Unscented Kalman filter-based adaptive tracking control for wheeled mobile robots in the presence of wheel slipping," in *Proc. 12th World Congr. Intell. Control Automat. (WCICA)*, Jun. 2016, pp. 3335–3340.
- [11] A. Onat, P. Voltr, and M. Lata, "A new friction condition identification approach for wheel–rail interface," *Int. J. Rail Transp.*, vol. 5, no. 3, pp. 127–144, 2017.
- [12] A. Onat, P. Voltr, and M. Lata, "An unscented Kalman filter-based rolling radius estimation methodology for railway vehicles with traction," *Proc. Inst. Mech. Eng., F, J. Rail Rapid Transit*, vol. 232, no. 6, pp. 1686–1702, 2018.
- [13] A. Nikoofard, U. J. F. Aarsnes, T. A. Johansen, and G.-O. Kaasa, "State and parameter estimation of a drift-flux model for underbalanced drilling operations," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 6, pp. 2000–2009, Nov. 2017.
- [14] Y. Wang, Z. Qiu, and X. Qu, "An improved unscented Kalman filter for discrete nonlinear systems with random parameters," *Discrete Dyn. Nature Soc.*, vol. 2017, Feb. 2017, Art. no. 7905690.
- [15] L. Escuin-Poole, J. Garcia-Ojalvo, and A. J. Pons. (2017). "Extracranial estimation of neural mass model parameters using the unscented Kalman filter." [Online]. Available: <https://arxiv.org/abs/1708.05282>
- [16] M. Witczak, "Fault diagnosis and fault-tolerant control strategies for non-linear systems," in *Lecture Notes in Electrical Engineering*, vol. 266. Cham, Switzerland: Springer, 2014, pp. 375–392.
- [17] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.
- [18] R. Van der Merwe and E. A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 6, May 2001, pp. 3461–3464.
- [19] E. Wan and R. Van Der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks*. Hoboken, NJ, USA: Wiley, 2001, ch. 7, pp. 221–280.
- [20] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adapt. Syst. Signal Process., Commun., Control Symp.*, Oct. 2000, pp. 153–158.
- [21] A. Yu, Y. Liu, J. Zhu, and Z. Dong, "An improved dual unscented Kalman filter for state and parameter estimation," *Asian J. Control*, vol. 18, no. 4, pp. 1427–1440, 2016.
- [22] B. Gao, S. Gao, G. Hu, Y. Zhong, and C. Gu, "Maximum likelihood principle and moving horizon estimation based adaptive unscented Kalman filter," *Aerosp. Sci. Technol.*, vol. 73, pp. 184–196, Feb. 2018.
- [23] B. Matzuka, M. Aoi, A. Attarian, and H. T. Tran, "Nonlinear filtering methodologies for parameter estimation," North Carolina State Univ., Raleigh, NC, USA, Center Res. Sci. Comput., Tech. Rep. CRSC-TR12-15, 2012.
- [24] R. Kandepu, B. Foss, and L. Imsland, "Applying the unscented Kalman filter for nonlinear state estimation," *J. Process Control*, vol. 18, nos. 7–8, pp. 753–768, 2008.
- [25] H. K. Khalil, *Nonlinear Systems*, vol. 3. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [26] E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmos. Sci.*, vol. 20, no. 2, pp. 130–141, 1963.
- [27] J. H. Gove and D. Y. Hollinger, "Application of a dual unscented Kalman filter for simultaneous state and parameter estimation in problems of surface-atmosphere exchange," *J. Geophys. Res., Atmos.*, vol. 111, no. D8, pp. 1–21, 2006.
- [28] J. Han, Q. Song, and Y. He, "Adaptive unscented Kalman filter and its applications in nonlinear control," in *Kalman Filter Recent Advances and Applications*. London, U.K.: IntechOpen, 2009.



ALTAN ONAT received the B.Sc. degrees in electrical–electronics and mechanical engineering and the M.Sc. degree in electrical–electronics engineering from Eskişehir Osmangazi University, Turkey, in 2009, 2010, and 2012, respectively, and the Ph.D. degree in transport means and diagnostics from the University of Pardubice, Pardubice, Czech Republic, in 2017.

He is currently a Researcher with Eskişehir Technical University. His research interests include state and parameter estimation for dynamic systems (especially for railway vehicles), railway vehicle dynamics, condition monitoring, and adaptive control applications.

...